

OHJELMISTOPROJEKTIHALLINNAN KEHITTÄMINEN SCRUM-MENETELMÄLLÄ

Panu Vuori

Opinnäytetyö
Kesäkuu 2014

Automaatioteknologian koulutusohjelma YAMK
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) VUORI, Panu	Julkaisun laji Opinnäytetyö	Päivämäärä 6.6.2014
	Sivumäärä 76	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty (X)
Työn nimi OHJELMISTOPROJEKTINHALLINNAN KEHITTÄMINEN SCRUM-MENETELMÄLLÄ		
Koulutusohjelma Automaatioteknologian koulutusohjelma. Ylempi ammattikorkeakoulututkinto.		
Työn ohjaaja(t) SELOSMAA, Seppo		
Toimeksiantaja(t) Metsys Oy VÄÄTÄINEN, Mika, Toimitusjohtaja		
<p>Tiivistelmä</p> <p>Perinteisesti ohjelmistoprojektit ovat toteutettu hyödyntäen 1970-luvulla kehitetty vesiputousmallia. Tässä mallissa projekti viedään lävitse lineaarisesti etukäteen määriteltyjen vaiheiden mukaisesti. Vaiheet ovat: esitutkimus, määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. Kun yksi vaihe on saatu päätökseen, voidaan siirtyä seuraavaan vaiheeseen, mutta myös paluu edelliseen vaiheeseen on mahdollinen.</p> <p>Teoriassa tämä malli toimii hyvin, mutta käytännössä tämä ei ole paras mahdollinen tapa toteuttaa ohjelmistoprojekteja. Ongelmia aiheuttaa mm. määrittelyvaiheessa tehdyn speksin muuttuminen projektin edetessä. Käytännössä on havaittu, että pysyviä vaatimusmäärittelyjä ei voida tehdä, koska osa vaatimuksista selviää vasta projektin aikana tai jo määritellyt vaatimukset muuttuvat kesken projektin.</p> <p>Edellä kuvattuihin ongelmiin on etsitty ratkaisua ns. ketteristä menetelmistä. Näissä menetelmissä arvostetaan yksilöä enemmän kuin prosesseja ja työkaluja, toimivaa ohjelmistoa enemmän kuin perusteellista dokumentaatiota, asiakasyhteistyötä enemmän kuin hankalia sopimusneuvotteluja ja muutoksiin reagoimista enemmän kuin suunnitelmien orjallista seuraamista. Kun perinteisessä vesiputousmallissa muutokset määrittelyvaiheessa tehtyyn speksiin aiheuttavat ongelmia ohjelmiston toteutusvaiheessa, niin ketterissä menetelmissä hyväksytään muutos ja ollaan valmiita reagoimaan niihin. Ketteriä menetelmiä on useita ja niistä yksi suosituimmista on Scrum-menetelmä.</p> <p>Opinnäytetyössä tutkitaan ohjelmistotalo Metsys Oy:n projektinhallinnan muutosta perinteisestä vesiputousmallista ketterään Scrum-menetelmään. Työssä seurataan Scrum-menetelmän toimivuutta noin puoli vuotta kestävässä pilottiprojektissa, mikä päättyy lopulta toimivan järjestelmän käyttöönottoon.</p>		
Avainsanat (asiasanat)		
Projektinhallinta, Scrum, ohjelmistokehitys, ketterä kehitys		
Muut tiedot		



Author(s) VUORI, Panu	Type of publication Master's Thesis	Date 6.6.2014
	Pages 76	Language Finnish
		Permission for web publication (X)
Title The usage of Scrum in software development project		
Master's degree Program Automation technology		
Tutor(s) SELOSMÄÄ, Seppo		
Assigned by Metsys Oy VÄÄTÄINEN, Mika, CEO		
<p>Abstract</p> <p>Traditionally, software development projects have been executed using by as call waterfall model. Waterfall model is a sequential design process where projects go through linearly, in straight line with beforehand defined phases. Phases are conception, initiation, analysis, construction, testing, production implementation and maintenance. Phases are executed one at time and when one phase is completed next phase can be started, return to the previous phase is also possible.</p> <p>In theory, this model works well. But in practice this is not the best way to follow through the project. In many cases the requirement specification changes during the project and waterfall model are not able to react with agility needed.</p> <p>For the above-described problem the agile software development method have been developed. In an agile development method the main values are individuals over the process and tools, working software over the comprehensive documentation, customer collaboration over the contract negotiation and responding to changes over following exactly the plan. Several agile development method are known and the one of the most popular is Scrum.</p> <p>The aim of this thesis was implement Scrum software development method for the Metsys company and find the answers a questions: how the transformation from the waterfall model to Scrum model has succeeded and whether the Scrum gives any advantage the development teams?</p>		
Keywords Agile, Scrum, project management, software development		
Miscellaneous		

SISÄLTÖ

1 JOHDANTO	5
2 OHJELMISTOTUOTANTO	7
2.1 Perinteinen ohjelmistoprojekti	7
2.2. Ongelmat ohjelmistotuotannossa	9
2.3 Ketterät menetelmät	11
2.3.1 Agile manifesti	11
2.3.1.1 Arvo 1	12
2.3.1.2 Arvo 2	12
2.3.1.3 Arvo 3	12
2.3.1.4 Arvo 4	13
2.3.2 Scrum-menetelmä	14
2.3.2.1 Empiirinen prosessin hallinta	14
2.3.2.2 Scrumin arvot	16
2.3.2.2.1 Sitoutuminen	16
2.3.2.2.2 Keskittyminen	16
2.3.2.2.3 Avoimuus	16
2.3.2.2.4 Kunnioitus	17
2.3.2.2.5 Rohkeus	17
2.3.2.3 Scrumin roolit	18
2.3.2.3.1 Tuotteen omistaja	18
2.3.2.3.2 Scrum-mestari	18
2.3.2.3.3 Kehitystiimi	18
2.3.2.4 Scrum-prosessi	19
2.3.2.4.1 Visiointi	19
2.3.2.4.2 Tuotteen työlistan muodostaminen	20
2.3.2.4.3 Sprintin suunnittelu	20
2.3.2.4.4 Sprintti	21
2.3.2.4.5 Päiväpalaveri (Daily Scrum)	22
2.3.2.4.6 Sprintin katselmointi	22
2.3.2.4.7 Sprintin jälkitarkastelu (Retrospektiivi)	22
2.3.2.5 Scrumin dokumentit	23
2.3.2.5.1 Tuotteen työlista	23
2.3.2.4.2 Julkaisun edistymiskäyrä	24

2.3.2.4.3 Sprintin työlista	24
2.3.2.4.4 Sprintin edistymiskäyrä.....	25
2.3.3 Muut ketterät menetelmät.....	25
2.3.3.1 XP (eXtreme Programming)	25
2.3.3.2 Adaptive Software Development (ASD).....	28
2.3.3.3 Crystal metodit	29
3 LÄHTÖKOHDAT PROJEKTIHALLINAN KEHITTÄMISELLE ..	31
3.1 Metsys Oy	31
3.2 Metsysin projektinhallinta.....	31
3.3 Nykyisen projektinhallinnan kartoitus	33
3.3.1 Hyvät puolet nykyisissä menetelmissä.....	33
3.3.2 Haasteet nykyisissä menetelmissä.....	34
3.3.2.1 Määrittelyvaihe.....	34
3.3.2.2 Toteutusvaihe	34
3.3.2.3 Testaus.....	35
3.3.2.4 Ylläpito.....	35
3.3.3 Kvantitatiivisen-kyselyn tulokset.....	35
3.3.3.1 Ensimmäinen kysely ennen Scrum-projektia.....	36
3.3.3.1 Toinen kysely projektin päättyttyä	36
3.4 Scrumin soveltuminen Metsysin käytäntöihin	36
3.5 Scrum-koulutus	37
4 TOTEUTUS.....	38
4.1 Wattensin projekti	38
4.2 Projektin aloitus.....	38
4.3 Projektin eteneminen.....	40
4.3.1 Sprintti 1 (5.1.2011–19.1.2011)	40
4.3.2 Sprintti 2 (19.1.2011 – 31.1.2011)	42
4.3.3 Sprintti 3 (31.1.2011–14.2.2011)	42
4.3.4 Sprintti 4 (14.2.2011 – 28.2.2011)	43
4.3.5 Sprintti 5 (28.2.2011 – 14.3.2011)	44
4.3.6 Sprintti 6 (14.3.2011 – 28.3.2011)	45
4.3.7 Sprintti 7 (28.3.2011 – 11.4.2011)	46
4.3.8 Sprintti 8 (11.4.2011 – 25.4.2011)	48
4.3.9 Sprintti 9 (25.4.2011 – 9.5.2011)	49
4.3.10 Sprintti 10 (9.5.2011 – 23.5.2011)	49
4.3.11 Sprintti 11 (24.5.2011 – 23.5.2011)	50
4.4 Projektin päättäminen.....	51
5 TULOKSET.....	52

6 TULOSTEN ANALYSOINTI	54
6.1 Scrum-menetelmän edut.....	54
6.1.1 Sprintin suunnittelukokous.....	54
6.1.2 Päiväpalaveri	55
6.1.3 Sprintin katselmointi ja – jälkitarkastelu.....	56
6.1.4 Scrumin dokumentit	56
6.1.4 Scrumin muut hyödyt Metsysille	59
6.2 Scrum-menetelmän haasteet ja ongelmat.....	59
6.3 Scrumin toimivuus Metsysin projektissa	60
6.4 Vaikutus laatuun.....	61
6.5 Vaikutus kustannuksiin	62
7 POHDINTA.....	62
LÄHTEET	66
LIITTEET	68
Liite 1. Agile Manifesti	68
Liite 2. Ensimmäinen kysely	69
Liite 3. Toinen kysely (ennen Scrum-projektia)	69
Liite 4. Kolmas kysely (Scrum-projektin jälkeen)	72

KUVIOT

KUVIO 1. Vesiputousmallin prosessi.....	7
KUVIO 2. Scrum prosessikaavio.	19
KUVIO 3. ADS-menetelmän prosessi.	28
KUVIO 4. Crystal metodin kategoriat.	30
KUVIO 5. Sprintti 1 edistymiskäyrä.....	41
KUVIO 6. Sprintti 3 edistymiskäyrä.....	42
KUVIO 7. Sprintti 4 edistymiskäyrä.....	43
KUVIO 8. Sprintti 5 edistymiskäyrä.....	45
KUVIO 9. Sprintti 6 edistymiskäyrä.....	46
KUVIO 10. Sprintti 7 edistymiskäyrä.....	47
KUVIO 11. Sprintti 8 edistymiskäyrä.....	48
KUVIO 12. Sprintti 9 edistymiskäyrä.....	49
KUVIO 13. Sprintti 10 edistymiskäyrä.....	50
KUVIO 14. Sprintti 11 edistymiskäyrä.....	51

KUVIO 15. Julkaisun edistymiskäyrä.....	53
KUVIO 16. Tuotteen työlista.	57
KUVIO 17. Julkaisun edistymiskäyrä.....	57
KUVIO 18. Sprintin työlista.	58
KUVIO 19. Sprintin edistymiskäyrä.	58

1 JOHDANTO

Tässä opinnäytetyössä tarkastellaan Scrum-menetelmän soveltuvuutta ohjelmistotalo Metsys Oy:n ohjelmistoprojektien hallinnassa. Opinnäytetyössä vietiin läpi Metsysin ensimmäinen projekti käyttäen Scrum-menetelmää. Työssä analysoidaan tämän projektin onnistumista ja soveltumista Metsysin toimintaympäristöön ja -tapoihin.

Scrum-menetelmä kuuluu ns. ketteriin menetelmiin, joita alettiin käyttämään ohjelmistoprojekteissa enemmässä määrin 1990-luvulla. Ketterien menetelmien pääpaino on toimivien ohjelmistojen tuottamisessa, suorassa viestinnässä ja nopeassa reagoinnissa muutoksiin. Ketterissä menetelmissä pyritään minimoimaan riski jakamalla ohjelmistokehitys lyhyisiin iteraatioihin, jotka kestävät tyypillisesti yhdestä neljään viikkoa. Kaikki ketterät menetelmät perustuvat Agile Manifestiin.

Haastaviksi ohjelmistoprojektit tekee määritysten puutteellisuus tai niiden muuttuminen kesken projektin. Uutta järjestelmää tilatessaan asiakas ei aina itsekkään tiedä yksityiskohtaisesti mitä ominaisuuksia uudelta järjestelmältä toivotaan, tästä seurauksena järjestelmän määritykset ja tarpeet muuttuvat läpi projektin.

Useasti Metsysin ohjelmistoprojekteissa on haasteita projektin läpinäkyvyydessä ja kommunikoinnissa projektiryhmän kesken. Lisäksi ohjelmistosuunnittelijoiden osallistuminen samanaikaisesti useaan projektiin luo haasteita projektien menestykselle läpiviemiselle. Näihin haasteisiin haettiin vastausta Scrum-menetelmästä.

Allekirjoittanut toimii pilottiprojektissa Scrum-mestarina, jonka vastuulla tässä projektissa on Scrum-menetelmien jalkauttaminen käytäntöön ja valvoa, että sovitut menetelmiä noudatetaan projektissa. Lisäksi projektiin osallistuu projektiohjaaja ja kolme ohjelmistosuunnittelijaa, projektiohjaaja osallistuu myös ohjelmointityöhön. Projektin päämääränä on tuottaa automaattitrukkien ohjausjärjestelmä paperitehtaalle Itävaltaan.

Luvussa kaksi kerrotaan tietoperusta perinteisistä menetelmistä viedä läpi ohjelmistoprojekti ja syvennyttään ketteriin-menetelmiin pääpainona Scrum-menetelmä.

Luvussa kolme analysoidaan Metsysin käytössä ollutta projektinhallintaa. Suoritimme kolmiosaisen mielipidekyselyn Metsysin henkilöstölle koskien käytössä olevaa projektinhallintaa ja sen toimivuutta. Kyselyn ensimmäisessä osiossa kysyttiin kaksi avointa kysymystä. Ensimmäisen kyselyn vastausten pohjalta suoritettiin kvantitatiivinen-kysely. Tämä kysely suoritettiin kahteen kertaan, ennen ja jälkeen pilottiprojektin.

Luvussa neljä kerrotaan pilottiprojektin etenemisestä sprintti kerrallaan. Alun perin olimme suunnitelleet, että projekti pystytään viemään lävitse kuudessa kahden viikon pituisessa sprintissä. Lopulta projekti pitkittyi erinäisistä syistä 11 sprintin mittaiseksi.

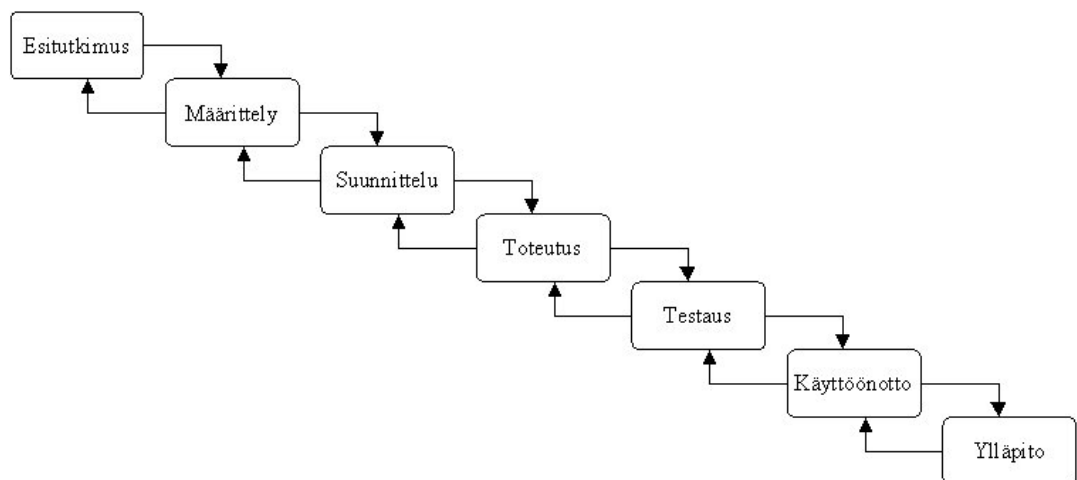
Luku viisi sisältää analysoinnin pilotti-projektista. Analysoin projektin onnistumista Scrum-menetelmän näkökulmasta ja tarkastelin Scrumin hyötyjä ja haasteita pilotti-projektissa Metsysin toimintaympäristössä.

Luvuissa kuusi ja seitsemän pohditaan Scrum-menetelmän sopivuutta Metsysin toimintaympäristöön ja mietitään miten menetelmää voidaan jatkossa hyödyntää entistä paremmin Metsysin projektinhallinnassa.

2 OHJELMISTOTUOTANTO

2.1 Perinteinen ohjelmistoprojekti

Perinteisesti ohjelmiston kehitystyö tai koko elinkaari on jaettu vaiheisiin ja siksi puhutaankin vaihejakomalleista. Perinteinen vaihejakomalli on vesiputousmalli (waterfall model), jonka yksi versio on esitetty kuviossa 1. 1970-luvulla kehitetty vesiputousmalli on näihin päiviin asti toiminut lähes aina ohjelmistokehitysprojektien pohjana. Vesiputousmallista on useita versioita vaiheineen, mutta yleisesti projektiin kuuluu ainakin määrittely, suunnittelu, toteutus, käyttöönotto ja ylläpito. Huolimatta vaiheistuksesta, kaikki vaiheet ovat riippuvaisia toisistaan. Vesiputousmallissa projekti etenee lineaarisena prosessina alusta loppuun. Toisinaan joudutaan palaamaan aiempaan vaiheeseen myöhemmin havaittujen seikkojen, kuten määrittelypuutteiden vuoksi. (Haikala & Märijärvi 2004, 36 - 37.)



KUVIO 1. Vesiputousmallin prosessi.

Esitutkimusvaihe alkaa yleisten, korkeamman tason vaatimusten selvittämisellä. Käytännössä tässä vaiheessa siis selvitetään miksi ohjelmisto tai järjestelmä tulee tehdä. Tämä vaihe on hyvin haastava, sillä asiakkaan todelliset tarpeet on saatava selville ja ne on ymmärrettävä perusteellisesti. Mikäli esitutkimusvaiheessa

määritetyt asiakasvaatimukset ovat virheellisiä tai väärin ymmärretty, ei hyvään lopputulokseen ole mahdollista päästä. (Haikala & Märijärvi 2004, 37.)

Määrittelyvaiheessa laaditaan ensin esitutkimusvaiheen tulosten perusteella vaatimusmäärittelydokumentti eli speksi, johon kirjataan kaikki asiakkaan tarvitsemat ominaisuudet. Tämäkin työvaihe vaatii äärimmäistä tarkkuutta, sillä juuri tätä dokumenttia vasten tarkastellaan projektin lopuksi sisältääkö lopputuote kaikki tilatut ominaisuudet. Määrittelyvaiheessa työestetään vielä vaatimusmäärittelyn pohjalta ohjelmiston toteutusmäärittelydokumentti, jossa kuvataan ohjelmiston toiminnot sekä rajoitukset. Toteutusmäärittelydokumenttiin voidaan kirjata myös sellaisia ohjelmistoa koskevia vaatimuksia, mitkä eivät ole toiminnallisia vaatimuksia, esimerkiksi suoritustehoon liittyviä vaatimuksia. (Haikala & Märijärvi 2004, 38-39.)

Suunnitteluvaiheessa suunnitellaan ohjelmiston toteutus toteutusmäärittelyn perusteella. Usein suunnitteluvaiheessa laaditaan ensin korkean tason suunnitelma (arkkitehtuurisuunnitelma) ja sen jälkeen suunnitellaan toteutus tarkammalla tasolla. (Haikala & Märijärvi 2004, 28-29)

Toteutusvaiheessa kirjoitetaan itse ohjelmakoodi suunnitteludokumentin mukaisesti. Ohjelmointivaihe katsotaan päättyneeksi, kun kaikki vaatimusmäärittelyssä mainitut ominaisuudet on toteutettu ja lopullisesta ohjelmistosta saadaan virheetön käännös. (Haikala & Märijärvi 2004, 40.)

Testaus on oma osaamisalueensa. Projektipäällikön täytyy ymmärtää testaamisen merkitys, koska jokaisessa projektissa testataan ja sen aiheuttama työmäärä on suuri. Huono testaaminen tai sen laiminlyönti saattavat aiheuttaa projektin lopputuloksessa ikäviä yllätyksiä. Projektin alkuvaiheessa on hyvä suunnitella strategia testaamiselle tai ainakin määritellä projektin eri vaiheille omat testaukset ja niille tavoitteet. (Lehtimäki 2006, 170.)

Käyttöönotto vaiheessa kehitetty ohjelmisto luovutetaan asiakkaalle ja otetaan todelliseen käyttöön. Tässä vaiheessa ohjelmisto myös integroidaan tarvittaessa muihin ympäröiviin järjestelmiin rajapintojen kautta.

Ohjelmiston käyttöönoton jälkeen ohjelmisto siirtyy yleensä ylläpidon piiriin. Ylläpito on asiakkaan ongelmien ratkomista, esimerkiksi virheiden korjaamista, neuvontaa sekä parannusehdotuksien käsittelemistä. Ohjelmistotuotteiden tapauksessa tuotteella ei useinkaan ole varsinaista ylläpitovaihetta. Korjaukset, muutokset ja lisäykset toteutetaan projektissa, jossa toteutetaan tuotteen seuraava versio. (Haikala & Märijärvi 2004, 41.)

2.2. Ongelmat ohjelmistotuotannossa

Eräs ohjelmistotuotannon tunnetuimmista asiantuntijoista, Frederic Brooks, mainitsee artikkelissaan *No Silver Bullet* ohjelmistotuotannon erityispiirteitä verrattuna perinteisiin tekniikan aloihin. Näitä erityispiirteitä ovat mm. ohjelmistojen luontainen monimutkaisuus, näkymättömyys ja muunneltavuus. Haikala & Märijärvi lisäävät listaan vielä ainutkertaisuuden, menetelmien skaalautumattomuuden ja ohjelmistoihin perustuvien järjestelmien tietynlainen epäjatkuvuus.

Monimutkaisuus. Ohjelmistotuotteet ovat luonnostaan monimutkaisia. Koska ongelmat, joita ohjelmistot ratkovat, ovat monimutkaisia, myös ohjelmat ovat pakostakin monimutkaisia. Hyvällä ohjelmistosuunnittelulla monimutkaisuutta ei pystytä poistamaan – huonolla suunnittelulla sitä kylläkin pystytään lisäämään. Ohjelmistotuotteen monimutkaisuus liittyy usein sen kokoon: suuret järjestelmät ovat yleensä myös monimutkaisia. (Brooks 1986, 3.)

Näkymättömyys. Projektinhallinnan kannalta ohjelmistokehitystyön hankalin ongelma on näkymättömyys: ohjelmistotyön keskeneräisistä tuotoksista on hyvin hankala sanoa projektin valmiusastetta. Projektin saattaa esimerkiksi olla hyvin aikataulussa, vaikka puolet ajasta on jo käytetty eikä riviäkään ohjelmakoodia ole vielä kirjoitettu. Vastaavasti projektin saattaa olla pahasti myöhässä, vaikka vasta puolet ajasta on käytetty ja järjestelmätestaus on jo alkanut. Projektinhallinnassa onkin tärkeää pyrkiä tekemään projektin todellinen eteneminen näkyville mm. vaiheistamalla projektin tilanteeseen parhaiten sopivalla tavalla sekä erilaisten välietappien ja laadunvarmistustoimenpiteiden avulla. (Brooks 1986, 4.)

Muunnettavuus. Ohjelmistoja on ”helppo” muuttaa. Tämä aiheuttaa ohjelmistoihin kohdistuvia muutospaineita. On tavallista, että ohjelmistolle asetetut vaatimukset tarkentuvat tai niitä muutetaan jo kehitysaikana puhumattakaan ohjelmiston ylläpitoajasta. Myös kustannustekijät saattavat vaikuttaa muutostarpeisiin. (Brooks 1986, 4.)

Ainutkertaisuus. Ohjelmistoprojektit ovat yleensä ainutkertaisia siinä mielessä, että toisin kuin esimerkiksi rakennuksia tai siltojen rakennettaessa, samantapaista ohjelmaa ei ehkä ole tehty koskaan aikaisemmin. Tämä on seurausta mm. alan nopeasta kehityksestä, joka tuo sekä uusia sovelluksia että uutta teknologiaa jatkuvasti ohjelmistotekniikan piiriin. (Haikala & Märijärvi 2004, 30.)

Skaalautumattomuus. Aiemmin hyväksi havaitut menetelmät eivät välttämättä toimi projektin koon kasvaessa. Pienehkön ohjelman tapauksessa (esimerkiksi alle 10 000 riviä) jokainen projektiin osallistuvista omaa yleensä riittävän tarkan käsityksen koko käsillä olevasta ohjelmistosta, ja projekti saattaa hyvinkin onnistua erinomaisesti ilman dokumentaatiota, projektisuunnitelmaa ja kunnollista projektinohjausta. Ongelmien ilmaantuessa projekti saattaa olla pelastettavissa jopa yksittäisen henkilön ”sankarisuorituksella”. Vaikeuksiin ajautunutta suurta projektia ei voi pelastaa millään yksittäisellä uroteolla, eikä edes henkilökunnan lisääminen auta, päinvastoin, ns. Brooksin laki ja kokemus ovat osoittaneet, että ihmisten lisääminen myöhässä olevaan ohjelmistoprojektiin useimmiten myöhästyttää sitä vielä entisestään. (Haikala & Märijärvi 2004, 30.)

Epäjatkuvuus. Ohjelmistoihin perustuvien järjestelmien käyttäytyminen virhetilanteissa on usein epäjatkuvaa. Pieni vika perinteisellä teknologialla toteutetussa laitteessa ei tavallisesti lopeta kokonaan sen toimintaa. Ohjelmistolle taas on tyypillistä, että mikä tahansa vähäpätöiseltäkin vaikuttava vika voi sekunnin murto-osassa saattaa koko järjestelmän käyttökelvottomaksi. Epäjatkuvuudesta johtuen ohjelmien toimintaa ei voida varmistaa testaamalla, koska kaikkia erikoistilanteita ja niiden yhdistelmiä ei mitenkään voida käydä testauksessa läpi. (Haikala & Märijärvi 2004, 30 - 31.)

2.3 Ketterät menetelmät

Ketterät ohjelmistokehityksen menetelmät tulivat yleisempään käyttöön 1990-luvulla. Perinteisen ohjelmistosuunnittelun – ja tuotannon mallit ja menetelmät eivät kyenneet kunnolla vastaamaan nopeasti kasvavan kehitystyön tarpeisiin. (Abrahamsson, ym. 2003,1)

Ketterien menetelmien keskeisiä ominaisuuksia ovat nopeus ja yksinkertaisuus. Perusideana on toteuttaa sovellukseen vain keskeisimmät (ja välittömästi tarvittavat) ominaisuudet. Tämän jälkeen ensimmäinen versio sovelluksesta julkaistaan ja kootaan palaute. Palautteen pohjalta sovellukseen tehdään pieniä muutoksia. Tarkoituksena on kehittää sovellusta pienin askelin eteenpäin lisäten uusia ominaisuuksia ja pyrkiä varsin tiiviissä tahdissa julkaisemaan uusia versioita sovelluksesta palautteen kokoamista varten. Tämän inkrementaalisen kehitystyön tavoitteena on pystyä reagoimaan nopeasti liike-elämän tarpeisiin ja teknologian muutoksiin. Keskeistä on siis itaratiivisuus, jolloin suunnittelu-, määrittely-, toteutus- ja testausvaiheet limittyvät suunnittelu- ja toteutusprosessissa. (Abrahamsson, ym. 2003,1)

Ketterät menetelmät pitävät suoraa viestintää (mieluiten kasvokkain) tärkeämpänä kuin kirjoitettuja dokumentteja. Useimmat ketterät tiimit työskentelevät samassa työtilassa, ja tiimiin kuuluvat kaikki, joita tarvitaan ohjelmiston saamiseen valmiiksi. Tämä tarkoittaa vähintään ohjelmoijia ja heidän ”asiakkaitaan”. (Asiakkaat määrittelevät tuotteen ja voivat olla tuotepäälliköitä, liiketoiminta-analyytikkoja tai varsinaisia käyttäjiä.) Tiimiin voi kuulua myös testaajia, käyttöliittymäsuunnittelijoita, teknisiä kirjoittajia ja päälliköitä. (Wikipedia, 2014)

2.3.1 Agile manifesti

Vuonna 2001 seitsemäntoista merkittävää ketterän menetelmän puolestapuhujaa kokoontui Utahissa keskustelemaan menetelmiensä yhteisestä perustasta. Tarkoitus oli luoda yhteistä pohjaa ketterille menetelmille ja edistää näin ketterän ajattelun leviämistä. Tuon kokoontumisen tuloksena he julkaisivat julistuksen nimeltä Agile Manifesto (LIITE 1), jota pidetään ketterän kehityksen perusmääritelmänä.

Manifestissa määritellään ketterille menetelmille neljä ydinarvoa, joita menetelmät noudattavat.

2.3.1.1 Arvo 1

Arvostetaan yksilöitä ja vuorovaikutustaitoja enemmän kuin prosesseja ja työkaluja.

Ihmiset tekevät yhdessä töitä valmistaakseen ohjelmia, ketterä-kehitys asettaa suuren arvon näille ammattilaisille, jotka muodostavat projektiryhmän. Lisäksi, että projekti onnistuisi, ryhmälle tulee valita oikeat prosessit ja työkalut, jotka sopivat heidän ympäristöönsä, teknologioihin ja projektin tavoitteisiin. Projektiryhmän projekti väärillä prosesseilla ja työkaluilla voi epäonnistua. (Schuh 2005,3.)

2.3.1.2 Arvo 2

Arvostetaan toimivaa ohjelmaa enemmän kuin perusteellista dokumentaatiota.

Kehitettävän ohjelman on tarkoitus antaa käyttäjälleen työn suorittamiseen lisäarvoa. Ketterän-kehityksen mielestä ei ole hyötyä hyvin suunnitelluista yksityiskohtaisista vaatimuksista, arkkitehtuurista ym. suunnitteludokumenteista, jos niitä ei pystytä yhdistämään toimivaan järjestelmään. Toimivan järjestelmän tulisi olla projektin toiminnan päätarkoituksena. Lisäksi liiallinen dokumentointi projektin aikaisessa vaiheessa voi olla hukkaan heitettyä aikaa, jos sitä ei ylläpidetä myöhemmin. Jos dokumentoinnista ja sen ylläpitämisestä tulee toiminnan päätarkoitus, voi se aiheuttaa projektin myöhästymisen. (Schuh 2005, 3-4)

2.3.1.3 Arvo 3

Arvostetaan yhteistyötä asiakkaan kanssa enemmän kuin hankalia sopimusneuvotteluja.

Ketterä-kehitys uskoo, että yhteistyö asiakkaan kanssa pitää projektin oikeilla raiteilla. Liian usein prosessit, työkalut ja dokumentaatio voivat aiheuttaa sen, että projektin päämäärä häviää projektintekijöiltä. Agile projektiryhmillä on käytäntöjä, jotka ohjaavat niitä olemaan avoimia asiakkaan tarpeille. Lisäksi Agile edistääkin

sopimussuhteita siten, että se rohkaisee työskentelemään asiakkaan kanssa ja asiakkaalle. (Schuh 2005, 4)

2.3.1.4 Arvo 4

Arvostetaan muutoksiin vastaamista enemmän kuin suunnitelmien orjallista seuraamista.

Muutoksia tapahtuu jokaisessa tietojärjestelmäprojektissa ja jokaisen tulee muuntaa itsensä työskentelemään muutosten parissa, koska muuten toimitetaan ala-arvoista tuotetta asiakkaan näkökulmasta katsottuna. Ketterä-kehitys tuo tähän käytäntöjä jotka ohjaavat tiimit mukautumaan projektissa muuttuviin vaatimuksiin ja ympäristötekijöihin. Lisäksi Agile kannustaa siihen, että projektiryhmä voi tarvittaessa räätälöidä prosesseja, suunnitelmia, dokumentaatiota ja sopimuksia saavuttaakseen päämäärän. (Schuh 2005, 4)

2.3.2 Scrum-menetelmä

Scrum on yksi ketteristä menetelmistä joka pohjautuu Agile Manifestoon. Terminä Scrum tulee urheilusta ja tarkemmin Rugbystä, molemmissa yhteyksissä Scrumilla tarkoitetaan itseohjautuvan joukkueen tai ryhmän sopeutumista erilaisiin ja muuttuviin tilanteisiin. Scrumin kehittäjinä pidetään Jeff Sutherlandia, Ken Schwaberiä, John Scumnotalesia ja Jeff McKennaa.

Scrumia on hyödynnetty monimutkaisten tuotteiden kehittämisessä 1990-luvun alusta lähtien eli jo ennen Agile Manifestoa. Scrum ei ole tuotekehitysprosessi tai -tekniikka, vaan paremminkin viitekehys, jonka sisällä voi käyttää useita erilaisia prosesseja ja tekniikoita. (Schwaber, Surherland 2013, 3)

Scrum tekee tuotehallinnon ja -kehityksen menetelmien vaikutukset näkyviksi, jotta menetelmiä voidaan parantaa. Scrum koostuu scrum-tiimeistä rooleineen, tapahtumista, tuotoksista ja säännöistä. Jokainen elementti palvelee tiettyä tarkoitusta ja on tärkeä osa Scrumin onnistumista. Scrumin säännöt sitovat yhteen roolit, tapahtumat ja tuotokset ja ohjaavat niiden välistä vuorovaikutusta. (Schwaber, Surherland 2013, 3)

Scrumin kehittäjät Schwaber ja Beedle (2004) ovat todenneet Scrumin olevan kevytrakenteinen ketterä projektinhallintamenetelmä, perustuen pienikokoiseen, päätäntävaltaiseen ja itsestään organisoituvaan tiimiin. Scrum-projektissa pyritään jokaisen iteraation (sprintin) aikana saada luotua valmista toiminnallisuutta asiakkaalle. Tarkoituksena on tunnistaa nopeasti projektia hidastavat esteet, maksimoida yhteistyötä, parantaa kommunikointia, ohjailla prosessina kiinnostuksen ja tarpeen välistä kaaosta ja saada ihmiset tuntemaan hyvää mieltä omasta työstään. Scrum-tiimin saadessa itse päättää ja suunnitella miten toteuttaa haluttu toiminnallisuus on sitoutuminen tekemiseen hyvä ja onnistuminen todennäköistä. (Schwaber 2004, xii - xiii.)

2.3.2.1 Empiirinen prosessin hallinta

Perinteisesti ohjelmistoala on luonteeltaan täysin ennalta-arvaamaton ja monimutkainen rakenteeltaan. Ohjelmistosuunnittelijoiden osaaminen vaihtelee ja ohjelmistoala on jatkuvassa muutoksessa. Vaatimukset muuttuvat kesken projektin ja uusia tekniikoita tulee jatkuvasti käyttöön otettavaksi. (Leffingwell 2007, 45.)

Täysin lineaarisessa ohjelmistokehityksessä teknologia, henkilöt ja vaatimukset kulkevat käsi kädessä. Jos tuotteen vaatimukset ovat projektin toteuttajalle hyvin selkeät ja heillä on täydellinen näkemys siitä, kuinka kehitystyö tapahtuu käytössä olevien teknologioiden kautta, on työssä vain vähän häiriöitä ja arvaamattomuutta. Tällöin työ etenee lineaarisesti ja virheiden ja uudelleen tekemisen määrä jää pieneksi. Kun puolestaan vaatimukset eivät ole projektin toteuttajalle selkeät ja käytetään uutta teknologiaa, lisääntyy häiriö projektissa. Paine julkaista uusia tuotteita käyttäen uutta teknologiaa on aiheuttanut sen, että varmuus ja luotettavuus on vaihdettu kilpailukyvyn hakemiseen. Tämä tekijä saa aikaiseksi sen, että kehitystiimit työskentelevät uusien teknologioiden kanssa, vaatimukset muuttuvat jatkuvasti ja aikataulut ovat tiukkipia. (Schwaber & Beedle 2002, 91 - 93.)

Scrum tukeutuu empiirisen prosessinhallintaan, jossa perusolettamuksena on, että kaikki tehtävät ovat aavistamattomia. Se käyttää iteratiivis-inkrementaalista (toistavaa-lisäävää) lähestymistapaa ennustettavuuden optimoimiseen, riskien kontrolloimiseen ja muutoksiin reagoimiseen. Kyky muuntautua tilanteen mukaan ja jatkuvasti arvioida tilanteita uudestaan korostuu empiirisessä prosessissa. Seuranta ja palaute auttavat tunnistamaan ongelmat. Tällöin projektiryhmä pystyy kykyjensä mukaan toimimaan tilanteen mukaisesti jokaisessa arvaamattomassa tilanteessa. Empiirisessä lähestymisessä tärkeintä on läpinäkyvyys, sopeutuminen ja tarkastelu. Läpinäkyvyys havainnollistaa ongelmat, tarkastelulla arvioidaan projektin tuloksia ja sopeuttaminen auttaa muutosten hyväksymiseen läpinäkyvyyden ja tarkastelun kautta. (Leffingwell 2007, 45.)

2.3.2.2 Scrumin arvot

Scrum sisältää viisi perusarvoa, joihin sen toiminnot perustuvat. Perusarvot ovat sitoutuminen, keskittyminen, avoimuus, kunnioitus ja rohkeus. Nämä perusarvot saavat voimansa Scrum-projektin jäsenistä. Uudet teknologiat ja monimutkaiset vaatimukset saavat tiimit ajautumaan hankaliin tilanteisiin, joista niiden on noustava itse, ilman ulkopuolisen apua. Päämäärän saavuttaminen on kompromissien ja umpikujien takana, joten tiimin on oltava päämäärätietoinen ja sitoutunut velvoitteisiinsa. (Schwaber & Beedle 2002, 147.)

2.3.2.2.1 Sitoutuminen

Tiimin tehtävä on sitoutua yhteiseen päämäärään. Tiimi on oikeutettu saamaan kaiken tuen, jotta he voivat saavuttaa sitoumuksensa. Perinteisissä menetelmissä tiimille usein vain annetaan ulkoa käsin työtehtäviä ja pahimmillaan vielä määrätään, *miten* työ on tehtävä. Scrumissa tiimi itse *sitoutuu* toimittamaan seuraavan sprintin aikana määrätyt toiminnallisuudet. Samalla tiimi saa sprintin aikana vapauden toimia parhaaksi katsomallaan tavalla. Scrum-mestari ja ympäröivä organisaatio on omalta osaltaan sitoutuvat tukemaan tiimiä poistamalla heitä häiritseviä tekijöitä. Jotta tiimi voi sitoutua johonkin tavoitteeseen, on tämän tavoitteen oltava niin selkeä, että sprintin jälkeen voidaan todeta, onko tiimi onnistunut vai ei. (Schwaber & Beedle 2002, 148.)

2.3.2.2.2 Keskittyminen

Tee työsi hyvin. Keskity vain siihen, minkä olet luvannut tehdä äläkä huolehdi muusta. Jokaiselle sprintille asetetaan selkeä tavoite. Tämän tavoitteen saavuttaminen on tiimin tärkein tehtävä. Kaikki muut oheisaktiviteetit ovat toissijaisia. (Schwaber & Beedle 2002, 149 - 150.)

2.3.2.2.3 Avoimuus

On tärkeää, että kaikki projektiin liittyvät tiedot ovat näkyviä kaikille. Scrumissa kaikki on avointa: projektin ja sprintin työlistat ovat julkisia, ja myös päiväpalaveri on

avoinna kaikille kiinnostuneille (joskin vain tiimiin kuuluvat saavat puhua). Myös jokaisen sprintin tulokset esitellään julkisesti. (Schwaber & Beedle 2002, 151.)

2.3.2.2.4 Kunnioitus

Ihmisiä käsitellään yksilöinä heidän taustojensa ja kokemusten mukaan. On tärkeää kunnioittaa kutakin tiimin jäsentä juuri sellaisena kuin hän on. Tiimin itseorganisoiutuvuutta ja työrauhaa on myös kunnioitettava. Jokaisen tiimiläisen on kunnioitettava muita tekemällä jatkuvasti parhaansa, ja odotettava samaa myös muilta. (Schwaber & Beedle 2002, 152.)

2.3.2.2.5 Rohkeus

Ole rohkea sitoutumaan, toimimaan, olemaan avoin ja odota muilta kunnioitusta. Keskittyneesti ja avoimesti toimiminen voi joissain tilanteissa vaatia rohkeutta. Esimerkiksi omat virheet on myönnettävä heti ja äänekkäästi, jotta niistä voidaan oppia. Usein tiimi itse tietää, miten joku ongelma kannattaa ratkaista, mutta joku ulkopuolinen taho on eri mieltä. Tällöin vaatii suurta rohkeutta toimia, mahdollisesti jopa vastoin yrityksen virallista toimintatapaa. Tätä ei kuitenkaan pidä sotkea vastuuttomuuteen. (Schwaber & Beedle 2002, 153.)

2.3.2.3 Scrumin roolit

Vesiputousmallin mukaisissa projekteissa on yleensä ainakin määrittäjä, suunnittelija, ohjelmoija, testaaja ja projektipäällikkö. Projektipäällikköä lukuun ottamatta kussakin roolissa voi olla useampia henkilöitä ja toisaalta yksi henkilö voi joissain tapauksissa kuulua useampaankin rooliin.

Scrum-projektissa esiintyy vain kolme eri roolia: Tuotteen omistaja, Scrum-mestari ja tiimi.

2.3.2.3.1 Tuotteen omistaja

Tuotteen omistaja on henkilö, joka viime kädessä vastaa tuotteen ominaisuuksista, siis "omistaa" tuotteen. Tuotekehitysprojekteissa omistaja on tyypillisesti tuotepäällikkö, asiakasprojekteissa se voi olla asiakkaan edustaja tai toimittajan tekninen projektipäällikkö. Tuotteen omistajan tehtävänä on tehdä kaikki päätökset tuotteen ominaisuuksista ja toiminnallisuuksiin vaikuttavista seikoista. (Sininen meteoriitti, 2011)

2.3.2.3.2 Scrum-mestari

Scrum-mestarin tehtävänä on huolehtia siitä, että tiimi voi tehdä työtään optimaalisella tavalla. Tiimiläiset raportoivat päivittäin ongelmista, jotka hidastavat töiden etenemistä ja mestarin tehtävänä on ratkoa nämä ongelmat. Tämän lisäksi hän johtaa päivittäiset aamupalaverit ja vastaa siitä, että Scrumia noudatetaan oikein. (Sininen meteoriitti, 2011)

2.3.2.3.3 Kehitystiimi

Tiimiin kuuluvat kaikki henkilöt, jotka projektia ovat tekemässä. Tiimin sisältä ei erikseen nimitetä arkkitehteja, ohjelmoijia, testaajia tai käyttöliittymäsuunnittelijoita, vaan tiimiin kasataan henkilöitä, joilla on tarvittava osaaminen. Sitten tiimi yhdessä rakentaa tuotteen. Tällä halutaan korostaa kunkin tiimiläisen olevan projektin kannalta yhtä tärkeä ja että tiimi yhdessä vastaa tuotteen kaikista puolista, ei koskaan yksittäinen henkilö. Tiimin sisällä kaikki tekevät kaikkensa projektin edistämiseksi. Käytännössä eri ihmiset osaavat luonnollisesti eri asioita, ja on järkevää, että kukin tekee sitä, minkä osaa parhaiten. Olennaista kuitenkin on, että tiimi vastaa itse

2.3.2.4.2 Tuotteen työlistan muodostaminen

Vision jälkeen muodostetaan alustava tuotteen työlista, lista tuotteeseen tarvittavista ominaisuuksista. Ennen toteutuksen aloittamista tuotteen omistajan on priorisoitava ominaisuuslista. (Sininen meteoriitti, 2011)

2.3.2.4.3 Sprintin suunnittelu

Sprintin suunnittelukokous on yhden päivän mittainen työpaja, minkä ensimmäisellä puoliskolla valitaan tuotteen työlistasta seuraavan sprintin vaatimukset ja toinen puoli käytetään seuraavan sprintin valmisteluun.

Kokoukseen osallistuvat Scrum-mestari, tuotteen omistaja ja tiimi. Ulkopuolisia asiantuntijoita voidaan pyytää konsultoimaan joissain erityistietämystä vaativissa toimenpiteissä. Ulkopuolisia kuunteluoppilaita kokoukseen ei päästetä.

Tuotteen omistajan on priorisoitava tuotteen työlista ennen kokousta. Mikäli tuotteen työlista tai tuotteen omistajaa ei ole saatavilla kokoukseen, Scrum-mestarin on edustettava tuotteen omistajaa.

Tiimi päättää kuinka suuri osa tuotteen työlistasta voidaan valita seuraavaan sprinttiin.

Tuotteen työlistan analysointiin ei voida käyttää enemmän aikaa kuin kokouksen ensimmäinen on puolisko. Mikäli jotkut kohdat vaativat selvitystä, se on tehtävä sprintin aikana.

Suunnittelukokousta ei voida jakaa useammalle päivälle, eikä puoliskojen välissä pidetä mitään erityistä taukoa.

Työpajan jälkipuoliskolla tuotteen omistajan on henkilökohtaisesti oltava paikalla vastaamassa tiimin kysymyksiin.

Tiimi vastaa itsenäisesti siitä, miten sprintin työlistaksi valitut vaatimukset muutetaan julkaistavissa olevaksi tuotteen osaksi.

Työpajan lopputuotteena on seuraavan sprintin työlista. Työlistan ei tarvitse olla lopullinen, mutta sen on oltava riittävän hyvä, jotta sprintin toteuttaminen voidaan aloittaa. (Sininen meteoriitti, 2011)

2.3.2.4.4 Sprintti

Sprintin pituus on rajattu yhdestä viikosta kahteen kuukauteen. Tämä on sellainen aika, jossa tiimin on ehdittävä toteuttaa jotain merkittävää, mikä on julkaistavissa olevaa laatua. Tätä pidemmät työrupeamat alkavat jo vaatia tarkempaa suunnittelua ja dokumentaatiota työn tueksi. Myöskään tuotteen omistaja ja muut sidosryhmät eivät usein halua odottaa tätä kauempaa ilman että näkevät todellista edistymistä projektissa.

- Tiimi voi etsiä ulkopuolista apua sprintin aikana.
- Kukaan ei saa tyrkyttää tukeaan tiimille, eikä kommentoida tiimin tekemisiä sprintin aikana. Tiimi on täysin itseohjautuva.
- Tiimi sitoutuu tuotteen työlistaan sprintin suunnittelukokouksessa. Sprintin aikana tuotteen työlistan muuttaminen on kiellettyä.
- Jos ilmenee että sprintti ei ole toteuttamiskelpoinen, Scrum-mestari voi keskeyttää sen koska tahansa ja aloittaa uuden sprintin suunnittelukokouksen. Scrum-mestari voi tehdä tämän omasta aloitteestaan tai jos tiimi tai tuotteen omistaja pyytää sitä. Tyypillisiä syitä sprintin keskeyttämiselle on valitun teknologian osoittautuminen käyttökelvottomaksi, odottamattomat muutokset liiketoimintaympäristössä tai mikäli joku ulkopuolinen tekijä on häirinnyt tiimiä.
- Jos tiimi toteaa, ettei kykene suoriutumaan sprintistä, se voi neuvotella tuotteen omistajan kanssa, mitkä ominaisuudet voidaan siirtää seuraaviin sprintteihin. Jos liian suuri osa sprintistä näyttää jäävän valmistumatta, Scrum-mestari voi keskeyttää sprintin edellä kuvatusti.
- Jos tiimi toteaa että se ehtisi toteuttamaan enemmän kuin mihin oli sitoutunut, se voi neuvotella tuotteen omistajan kanssa mitä ominaisuuksia sprinttiin voidaan lisätä.
- Tiimiläisillä on kaksi hallinnollista velvollisuutta sprintin aikana: heidän on osallistuttava päiväpalaveriin ja pidettävä sprintin työlista ajan tasalla. Sprintin

työlistan on oltava julkisesti nähtävillä ja arviot jäljellä olevasta työmäärästä on päivitettävä päivittäin. (Sininen meteoriitti, 2011)

2.3.2.4.5 Päiväpalaveri (Daily Scrum)

Joka päivä tiimi kokoontuu yhteen lyhyeen tilannepalaveriin, jossa kukin tiimin jäsen vastaa kolmeen kysymykseen:

1. Mitä teit edellisen päivän aikana?
2. Mitä aiot tehdä seuraavan päivän aikana?
3. Mitkä tekijät estävät (tai hidastavat) sinua saavuttamasta sprintin tavoitteita?

Palaveriin osallistuvat ainakin kaikki tiimin jäsenet ja scrum-mestari. Palaveri on myös avoin kaikille muille, jotka ovat jollain tavalla projektista kiinnostuneita. Vain tiimiläiset saavat kuitenkin puhua. Tapahtuman tarkoitus on nimenomaisesti tarjota kaikille tieto siitä, missä projekti menee ja mitä ongelmia sillä on. Jos jostain asiasta pitää keskustella tarkemmin, sitä varten pidetään oma palaverinsa, jossa ovat läsnä vain ne henkilöt, joita asia koskettaa. Päiväpalaverin suositeltava kesto on 15 minuuttia ja usein palaverin venähtämisen ehkäisemiseksi se pidetään seisten. (Sininen meteoriitti, 2011)

2.3.2.4.6 Sprintin katselmointi

Jokaisen sprintin lopuksi tiimi esittelee valmista tuotetta tuotteen omistajalle. Sprintin lopuksi tuotteen siis pitäisi olla periaatteessa käyttöön otettavissa: se on toteutettu, testattu, dokumentoitu, käyttöliittymä on valmis ja niin edelleen. Näin omistaja voi päättää joko seuraavan sprintin tekemistä tai tuotteen käyttöönottamisesta. Tähän tapahtumaan saa osallistua kuka tahansa, joka on projektista kiinnostunut. (Sininen meteoriitti, 2011)

2.3.2.4.7 Sprintin jälkitarkastelu (Retrospektiivi)

Kunkin sprintin lopuksi tiimi, Scrum-mestari ja omistaja tarkastelevat päättynyttä sprinttiä. Kukaan tiimin jäsen kertoo omasta näkökulmastaan, mikä sprintissä meni

hyvin ja mitä voisi parantaa. Lopuksi tiimi yhdessä priorisoi kehityskohteet ja pyrkii toteuttamaan muutokset seuraavan sprintin aikana. Sprintin jälkitarkastelun jälkeen kierros alkaa alusta uuden sprintin suunnittelulla. Tässä vaiheessa omistaja voi jälleen vapaasti muuttaa tuotteen työlistaa, ja tiimi arvioi taas uudelleen, minkä ominaisuuksien toteuttamiseen se voi sitoutua. (Sininen meteoriitti, 2011)

2.3.2.5 Scrumin dokumentit

Scrumin dokumentaatio pitää sisällään neljä dokumenttia, jotka ovat tuotteen työlista (Product Backlog), sprintin työlista (Sprint Backlog), julkaisun edistymiskäyrä (Release Burndown) ja sprintin edistymiskäyrä (Sprint Burndown).

2.3.2.5.1 Tuotteen työlista

Tuotteen työlistassa listataan kaikki vaatimukset tuotteelle, jota kehitystiimit(t) kehittävät. Tuoteomistaja vastaa tuotteen työlistasta, sen sisällöstä, saatavuudesta ja priorisoinnista. Tuotteen työlista ei ole koskaan valmis ja sen ensiversio sisältää ainoastaan tunnetut ja parhaiten ymmärretyt vaatimukset. Tuotteen työlista kehittyy yhdessä itse tuotteen ja sen käyttöympäristön kanssa. Tuotteen työlista on dynaaminen ja se muuttuu jatkuvasti sisältämään vaatimukset, jotka tuote tarvitsee ollakseen tarkoituksenmukainen, kilpailukykyinen ja hyödyllinen. Tuotteen työlista on olemassa yhtä kauan kuin tuote. (Schwaber, Surherland 2013, 12)

Tuotteen työlista sisältää kaiken tarpeellisen menestyvän tuotteen kehittämiseksi ja julkaisemiseksi. Tuotteen työlista on lista kaikista ominaisuuksista, toiminnoista, teknologioista, parannuksista ja korjattavista virheistä, jotka tullaan toteuttamaan tuleviin tuotejulkaisuihin. Jokaisella työlistan kohdalla on kuvaus, prioriteetti ja aika-arvio. Prioriteetti määritellään riskin, lisäarvon ja tarpeellisuuden perusteella. Tuotteen työlista järjestetään sen kohtien prioriteettien mukaan. Korkeimman prioriteetin kohdat ohjaavat seuraavaksi aloitettavaa kehitystä. Korkeampi prioriteetti tarkoittaa, että työlistan kohta on kiireellisempi, sitä on ehditty suunnitella enemmän ja sen arvosta vallitsee suurempi yksimielisyys. Korkeamman prioriteetin kohdat ovat selkeämpiä ja sisältävät tarkempaa tietoa kuin matalamman prioriteetin kohdat. Korkeamman prioriteetin kohtien suurempi selkeys kasvattaa myös aika-arvioiden

tarkkuutta. Mitä matalampi prioriteetti, sitä vähemmän työlistan kohdasta tiedetään ja sitä vaikeampaa kohtaa on ymmärtää. (Schwaber, Surherland 2013, 12)

2.3.2.4.2 Julkaisun edistymiskäyrä

Julkaisun edistymiskäyrä on diagrammi, joka esittää tuotteen työlistassa jäljellä olevan työmäärän suhteessa aikaan. Arvioitu työmäärä on missä tahansa yksikössä, jonka scrumtiimi ja organisaatio on valinnut. Tuotteen työlistan vaatimuksille annetaan alustavat aika-arviot julkaisun suunnittelun yhteydessä sekä sitä mukaa, kun uusia vaatimuksia lisätään tuotteen työlistaan. Aika-arviot tarkentuvat tuotteen työlistan säännöllisissä tarkasteluissa. Aika-arvioita voi muuttaa milloin tahansa muulloinkin. Aika-arvioiden antamisesta vastaa kehitystiimi. Tuoteomistaja voi vaikuttaa kehitystiimiin auttamalla sitä ymmärtämään ja valitsemaan kompromisseja, mutta lopullisen aika-arvion antaa aina kehitystiimi. Tuoteomistaja pitää ajantasaisen julkaisun edistymiskäyrän jatkuvasti näkyvillä. Jäljellä olevan työmäärän muutoksista voidaan myös piirtää trendiviiva. (Schwaber, Surherland 2013, 13)

2.3.2.4.3 Sprintin työlista

Sprintin työlista koostuu tehtävistä, jotka kehitystiimin tulee toteuttaa muuttaakseen tuotteen työlistasta valitut kohdat julkaisukelpoiseksi tuoteparannukseksi. Tehtävät määritellään useimmiten sprintin suunnittelukokouksessa. Ne kuvaavat työtä, jonka kehitystiimi tunnistaa tarpeelliseksi sprintin tavoitteeseen pääsemiseksi. Sprintin työlistaan valitut kohdat tulee hajottaa niin pieniksi, että sprintin edistyminen voidaan havaita päiväpalavereissa. Sprintin työlistan kohdan tyypillinen koko on yksi miestyöpäivä tai vähemmän. (Schwaber, Surherland 2013, 14)

Kehitystiimi ylläpitää ja päivittää sprintin työlistaa koko sprintin ajan. Aloittaessaan sprintin toteutustyön kehitystiimi voi havaita, että tarvitaankin vähemmän tai enemmän tehtäviä kuin suunniteltiin, tai että jokin tehtävä tulee viemään enemmän tai vähemmän aikaa kuin arvioitiin. Kehitystiimi lisää tarvittavat uudet tehtävät sprintin työlistaan. Kun tehtäviä työestetään tai ne valmistuvat, kunkin tehtävän arvioitu jäljellä oleva työmäärä päivitetään. Mikäli tehtävä havaitaan tarpeettomaksi, se poistetaan. Ainoastaan kehitystiimi voi muuttaa sprintin työlistaa, sen sisältöä tai aika-arvioita

sprintin aikana. Sprintin työlista on selkeästi näkyvä, reaaliaikainen kuva työstä, jonka kehitystiimi suunnittelee saavansa valmiiksi sprintin aikana. Tämän vuoksi sprintin työlista on täysin kehitystiimin omistuksessa ja hallinnassa. (Schwaber, Surherland 2013, 14)

2.3.2.4.4 Sprintin edistymiskäyrä

Sprintin edistymiskäyrä on diagrammi, joka esittää sprintin työlistassa jäljellä olevan työmäärän suhteessa aikaan. Sprintin edistymiskäyrä piirretään laskemalla yhteen kaikki sprintin työlistassa olevat aika-arviot ja päivittämällä tätä lukua kerran päivässä. Luku kuvaa sprintin koko jäljellä olevaa työmäärää päivän tarkkuudella. Piirtämällä viiva sprintin kokonaistyömäärää päivän tarkkuudella kuvaavien pisteiden läpi kehitystiimi näkee konkreettisesti etenemisensä kohti sprintin tehtävien valmistumista. Tehtäviin käytetyn työajan pituudella ei ole Scrumissa merkitystä. Ainoastaan jäljellä oleva työmäärä ja päivämäärä ovat kiinnostavia muuttujia. (Schwaber, Surherland 2013, 14)

2.3.3 Muut ketterät menetelmät

Ketteriä menetelmiä on olemassa kymmenkunta ja jokaisella on omat kannattajansa. Kaikilla menetelmillä on yhteinen perusmääritelmä, joka pohjautuu yhdessä sovittuun Agile Manifestoon. Seuraavassa on kerrottu lyhyesti muutamasta muusta ketterästä menetelmästä.

2.3.3.1 XP (eXtreme Programming)

Extreme Programming on nimensä mukaisesti hyvin ohjelmointikeskeinen menetelmä. Ensimmäinen XP-projekti suoritettiin maaliskuussa 1996. Siitä lähtien menetelmää on käytetty menestyksekkäästi ympäri maailmaa useissa erikokoisissa projekteissa eri teollisuusaloilla. (Asteins, Granville, Novak 2002, 1)

XP:n vahvuus on sen asiakastyytyvyyden korostaminen. Lisäksi menetelmä antaa ohjelmistokehittäjille mahdollisuuden reagoida nopeasti asiakkaan muuttuville vaatimuksille. Menetelmä korostaa tiimityöskentelyä. Johtajat, asiakkaat ja ohjelmistokehittäjät ovat kaikki tasa-arvoisia jäseniä yhteisessä tiimissä. Tiimit ovat itseorganisoituvia. XP:ssä toteutetaan yksinkertaista, mutta tehokasta työskentely-

ympäristöä, jonka avulla tiimit pystyvät työskentelemään erittäin tuotteliaasti. (Asteins & ym. 2002, 4-8)

XP tuo ohjelmistoprojekteihin viisi keskeistä ominaisuutta: viestintä, yksinkertaisuus, palaute, kunnioitus ja rohkeus. Ohjelmistokehittäjät ovat jatkuvasti yhteydenpidossa sekä asiakkaiden että kollegoiden kanssa, tämä parantaa viestinnän toimivuutta. Ohjelmakoodin rakenne pidetään yksinkertaisena ja selkeänä. Ohjelmoijat saavat hyödyllistä palautetta ohjelmastaan testaamalla ohjelmistoaan ensimmäisestä päivästä lähtien. Lisäksi asiakkaalle toimitetaan ohjelmisto niin aikaisessa vaiheessa, kuin on mahdollista ja saadun palautteen pohjalta tehdään ehdotetut muutokset. Jokaisen onnistuneen iteraation jälkeen kehittäjien kunnioitus omaan ja tiimin muiden jäsenten tekemiseen lisääntyy. Tältä pohjalta tiimin jäsenet pystyvät rohkeasti reagoimaan muuttuviin vaatimuksiin ja teknologiaan. (Asteins & ym. 2002, 4-8)

Extreme Programming-menetelmään sisältyy myös useita yksinkertaisia sääntöjä. Yksittäiset säännöt saattavat kuulostaa hankalilta ja jopa naiiveilta, mutta yhdessä säännöt muodostavat selkeät arvot ja periaatteet XP:lle. Säännöt on rakennettu tasapainottamaan toisiaan, joten on tärkeää, että sääntöjä noudatetaan yhdessä. Alla listattuna 14 sääntöä, mihin XP perustuu:

1. Asiakkaan edustaja työskentelee suoraan projektitiimin kanssa.
2. Käytä metaforia vaikeiden kokonaisuuksien kuvailemiseen.
3. Suunnittelu
4. Lyhyet palaverit
5. Testit ennen koodia
6. Yksinkertaisin mahdollinen rakenne ohjelmakoodiin
7. Pariohjelmointi
8. Kaikilla käytössä yhteiset ohjelmointistandardit
9. Ohjelmakoodin jaettu omistajuus
10. Integroikaa jatkuvasti
11. Refaktoroikaa
12. Julkaisut pienissä inkrementteissä
13. Varokaa loppuun palamista
14. Hyväksykää muutokset

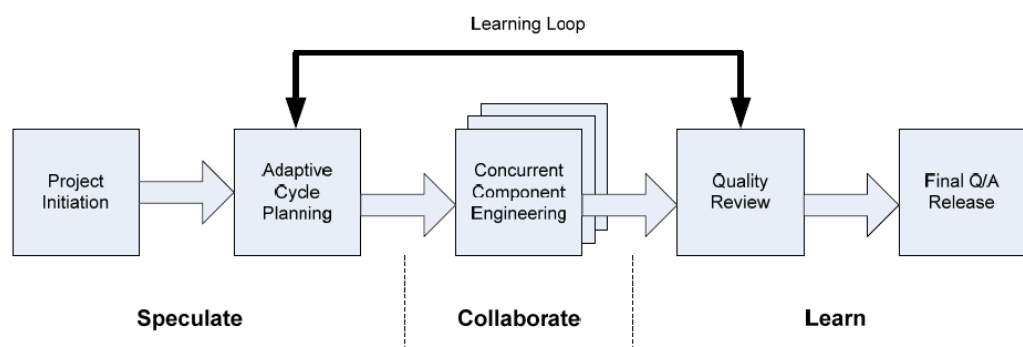
(Asteins & ym. 2002, 4-15)

2.3.3.2 Adaptive Software Development (ASD)

Adaptive Software Development (ASD) on suunnattu selkeästi projektin hallinnan kehittämiseen. Menetelmä ei ota kantaa ohjelmistokehitystekniikoihin, vaan niiden osalta kehitystiimille jätetään vapaat kädet. (Highsmith 2000, 70)

ASD:ssä, kuten muissakin agile-menetelmissä muutosta ohjelmistoprojektissa pidetään luonnollisena. ASD:ssä poikkeamat kesken projektin mielletään oikeaan suuntaan ohjaaviksi, positiivisiksi asioiksi. Perusajatuksena on, että alun perin suunniteltu järjestelmä ei ole projektin lopussa se, mitä toimeksiantaja lopulta haluaa. (Highsmith 2000, 43)

ASD:n prosessi muistuttaa tavoitteiltaan ja sisällöltään perinteistä vesiputousmallia. Erona vesiputousmalliin on kuitenkin se, että vaiheet voivat olla osittain samanaikaisia ja aikaisempiin vaiheisiin palaaminen on otettu huomioon. Lisäksi toteutusvaihe suoritetaan agile-menetelmille tutuissa iteraatioissa. ASD:n prosessi koostuu kuvion 3 mukaisesti kolmesta päävaiheesta: spekulointi (speculate), yhteistyö (collaborate) ja oppiminen (learn). Jokainen päävaihe sisältää tarkempia alivaiheita. (Highsmith 2000, 41,84)



KUVIO 3. ADS-menetelmän prosessi.

2.3.3.3 Crystal metodit

Alistar Cockburn ”julkaisi” Crystal-metodit vuonna 1997 ja ne olivat mukana, kun sovittiin ketteristä-menetelmistä Agile Manifestissä vuonna 2001.

Crystal-metodit ovat joukko projektinhallinnan käytäntöjä, joita voi soveltaa erilaisiin ja erikokoisiin ohjelmistoprojekteihin. Menetelmien lähtökohta on tarjota ihmislähtöisiä projektinhallinta malleja ohjelmistoprojektin suunnittelemiseksi. Crystal ei tarjoa valmiista prosessia, vaan ainoastaan joukon ohjeita ja sääntöjä, joiden perusteella jokainen projektiryhmä voi räätälöidä omiin tarkoituksiinsa sopivan prosessin. (Cockburn 2007, 335 - 337)

Kaikki Crystal-metodit noudattavat näitä Cockburnin määrittelemiä sääntöjä sekä prosessinsuunnitteluperiaatteita. Säännöissä mainitaan mm. että ohjelmistokehityksen on oltava inkrementaalista ja että prosessinarviointipalavereja on pidettävä ainakin jokaisen inkrementin lopussa ja alussa. (Cockburn 2007, 339)

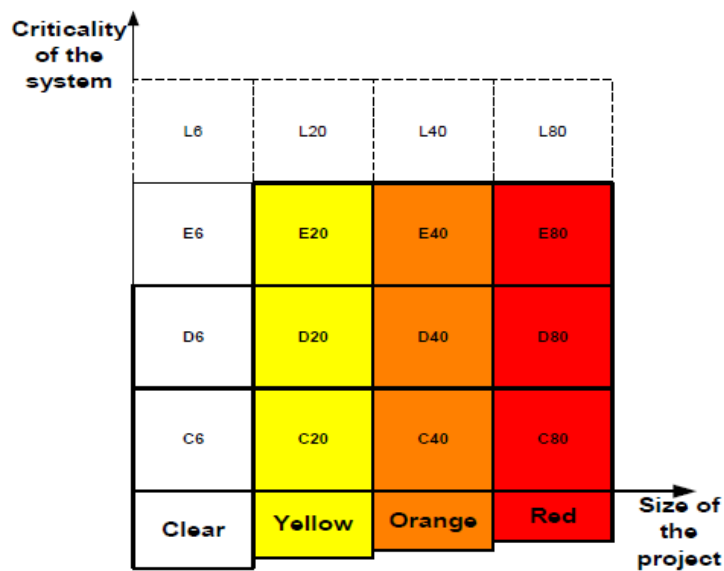
Cockburin seitsemän prosessinsuunnitteluperiaatetta on:

1. Mitä suuremmaksi tiimin koko kasvaa, sitä raskaampi ja kontrolloidumpi prosessin tulee olla.
 2. Kriittisemmät projektit vaativat tiukemman ja tarkemmin määritellyn prosessin.
 3. Kommunikointi kasvotusten on tehokkain kommunikointi tapa.
 4. Ylimääräiset vaiheet ja vaihetuotteet tulevat kalliiksi.
 5. Palautteen ja kommunikoinnin lisääminen vähentää vaihetuotteiden tarvetta.
 6. Omaksutut työskentelytavat, taidot ja ymmärtäminen mielummin, kuin prosessit, kaavamaisuus ja dokumentit.
 7. Tehokkuudesta voi tinkiä toiminna, jotka eivät voi muodostua pullonkaulaksi.
- (Cockburn 2007, 182)

Crystal-menetelmät ovat kategorioitu suoritettavan projektin laajuuden ja kriittisyyden mukaan (kuvio 4). Jos projektiin osallistuvia henkilöitä on kuusi tai alle, valitaan Crystal Clear. Henkilömäärän ollessa 7-20 on suositeltavaa valita Crystal Yellow. Kaikista suurimmissa projekteissa, 41–80 henkilöä on paras valinta Crystal Red. Mitä

suuremmaksi tiimin koko kasvaa, sitä raskaampi ja kontrolloidumpi prosessin tulee olla. (Cockburn 2007, 186)

Kriittisyydet ovat myös jaettu neljään kategoriaan viasta koituvien mahdollisten seurausten mukaan. Lievin kategoria on mukavuuden menettäminen (C = Comfort). Seuraavat kategoriat ovat taloudellinen tappio (D = Discretionary money) ja merkittävä taloudellinen tappio (E = Essential money). Äärimmäisen kriittisyyden kategoria on vika, jonka seurauksena voi olla kuolema (L = Life). Kriittisemmät projektit vaativat tiukemman ja tarkemmin määritellyn prosessin. (Cockburn 2007, 338)



KUVIO 4. Crystal metodin kategoriat.

3 LÄHTÖKOHDAT PROJEKTINHALLINAN KEHITTÄMISELLE

Metsysin ohjelmistokehitysprojektit toteutetaan pääsääntöisesti yhteistyössä Rocla Oyj:n kanssa. Metsys on kehittänyt MetRo-varastohallintajärjestelmän, joka tarvittaessa pystytään integroimaan Roclan automaattitrukki-järjestelmään. Loppuasiakkaalle toimitetaan heidän halutessaan järjestelmä, mihin on integroitu nämä kaksi järjestelmää. Integroidulla järjestelmällä ohjataan automaattisesti toimivia miehittämättömiä AGV (Automated Guided Vehicle)-vaunuja, vaunut suorittavat itsenäisesti haluttuja tehtäviä. Loppuasiakkaina toimivat eri prosessiteollisuuden toimijat, esimerkiksi paperitehtaat. Roclan AGV-järjestelmä pystyy toimimaan myös itsenäisesti ilman MetRo-järjestelmää, mutta integroiduissa järjestelmässä pystytään AGV-vaunuja ohjaamaan monipuolisemmin ja fiksummin juuri loppuasiakkaan tarpeiden mukaan. Lisäksi AGV-järjestelmän ylläpitäminen ja seuraaminen on käyttäjäystävällisempää ja monipuolisempaa MetRo-järjestelmän kautta.

3.1 Metsys Oy

Metsys on asiantuntija organisaatio, jossa koko henkilöstöllä on laaja osaaminen teollisuuden tietojärjestelmistä, joita yritys on kehittänyt jo yli 20 vuotta. Metsysin juuret ovat paperiteollisuudessa ja Kankaan paperitehtaalla. Kankaalle Metsys toimitti erinäisiä järjestelmiä eri tarpeisiin, järjestelmiä oli mm. tilausten käsittely, materiaalin hallinta, valitusten käsittely, henkilöstönhallinta ja tuotannon seuranta. Kankaan lisäksi Metsys on toimittanut järjestelmiä useisiin paperi – ja sellutehtaisiin ympäri maailman.

Vuonna 2007 Metsys teki ensimmäisen projektin Roclan kanssa ja siitä lähtien yhteisiä projekteja on ollut säännöllisesti. Näissä projekteissa Rocla tekee toimitussopimuksen loppuasiakkaan kanssa ja Metsys solmii alihankkijasopimuksen Roclan kanssa.

3.2 Metsysin projektinhallinta

Metsysillä ei ole ollut käytössä mitään selkeää projektinhallintamallia. Jokainen projekti on viety lävitse totutuilla tavoilla ja ohjelmistokehittäjä vetoisesti. Voidaan sanoa, että projektit on viety lävitse löyhästi vesiputous-mallia noudattaen eli määrittely, toteutus, käyttöönotto ja ylläpito.

Järjestelmän määrittely alkaa jo tarjouksen tekemisen yhteydessä, koska jotta voit tehdä järjestelmän toteutuksesta tarjouksen, tulee sinun tuntea mahdollisimman tarkasti tulevan projektin toiminnallisuus. Mahdollisen kaupan varmistuttua Metsysin edustaja käy Roclan edustajan kanssa tutustumassa loppuasiakkaan tuotantolaitokseen ja sen tuotantoprosesseihin. Samalla järjestetään aloituspalaveri missä tulevan projektin speksit tarkentuvat, näillä lähtötiedoilla järjestelmää aloitetaan kehittämään.

Projektin toteutusvaiheessa projektilla ei ole nimettyä projektipäällikköä vaan kehitysprosessi viedään läpi projektiin osallistuvien ohjelmistokehittäjien johdolla, yhdellä ohjelmistokehittäjällä on kuitenkin päävastuu projektista ja hän toimii projektin vastuuhenkilönä. Asiantuntijat sopivat itseohjautuvasti projektikäytännöistä työkuorman, erityisosaamisen ja vakiintuneiden työtapojen perusteella. Yrityksen johto ei ota aktiivisesti kantaa projektien henkilöstövalintoihin. Projektin vastuuhenkilö hoitaa pääasiallisen yhteydenpidon asiakasrajapinnassa.

Testauksessa ei ole ollut käytössä mitään systemaattista prosessia vaan järjestelmää on testattu asiantuntijoiden toimesta sitä mukaan kun järjestelmää on kehitetty. Projektin aikana on järjestetty yksi tai kaksi isompaa testipäivää joihin myös Roclan asiantuntijat ovat osallistuneet. Nämä testipäivät on todettu erittäin toimiviksi ratkaisuiksi. Järjestelmiä mitkä tehdään Roclan kanssa yhteistyössä päästään testaamaan Roclan tarjoaman AGV-simulaattorin avulla.

Käyttöönoton alkaessa järjestelmän pitää olla määritysten mukainen ja kattavasti testattu. Samanaikaisesti MetRo-järjestelmän kanssa käyttöönotetaan myös Roclan AGV-vaunut, tämä on erittäin intensiivinen vaihe ja kaiken pitää toimia mahdollisimman hyvin. Loppuasiakkaan tuotanto ei saa keskeytyä käyttöönoton takia, joten paineet saada järjestelmät toimimaan ovat suuret. Esimerkiksi paperitehtaan automatisoitu rullavaraston ei tarvitse olla pitkään poissa käytössä, kun tuotanto paperikoneilla jo pysähtyy. Käyttöönotto toteutetaan lähes poikkeuksetta paikanpäällä loppuasiakkaan luona. Joitakin pienempiä järjestelmiä on käyttöönotettu etäyhteyden

avulla toimistolta käsin. Kun hektisin vaihe käyttöönotosta on takanapäin alkaa useasti loppuasiakkaalta tulemaan muutospyyntöjä MetRo-järjestelmään. Halu muutoksille tulee, kun loppuasiakas näkee toiminnassa olevasta järjestelmästä mitkä toiminnallisuudet ovat järkeviä ja mitkä voitaisiin tehdä vielä paremmin, jotta tuotantoprosessit olisivat mahdollisimman tehokkaita.

Hyväksyntä järjestelmän toimittamisesta tulee, kun kaikki toimitussopimuksen ehdot täyttyvät. Tämän jälkeen järjestelmä siirtyy ylläpidon piiriin. Osa loppuasiakkaista ottaa laajennetun ylläpidon, joka tarkoittaa sovitun mukaista järjestelmän päivystämistä ja reagoimista ongelmatilanteisiin. Metsysillä on 24/7 asiantuntija tavoitettavissa näille loppuasiakkaille jotka ovat ottaneet laajennetun ylläpidon.

3.3 Nykyisen projektinhallinnan kartoitus

Henkilöstölle suoritettiin kolmiosainen mielipidekysely, jossa kartoitettiin nykyisen projektinhallinnan hyviä, huonoja ja kehitettäviä ominaisuuksia. Kyselyyn osallistui viisi henkilöä. Kyselyn ensimmäinen osio suoritettiin kahdella avoimella kysymyksellä. Avoimilla kysymyksillä haluttiin vastaukset nykyisen projektinhallinnan hyviin ja huonoihin puoliin. Ensimmäisen kyselyn vastausten pohjalta suoritettiin kvantitatiivinen-kysely, kyselyyn annettiin vastaukset asteikolla 0 – 10. Kvantitatiivinen-kysely suoritettiin uudelleen keväällä Scrum-projektin jälkeen. Kaikki kyselyt suoritettiin sähköpostin avulla. Kyselylomakkeet löytyvät liitteistä 1-3.

Ensimmäisenä toteutetun kyselyn kysymykset jätettiin avoimiksi, jotta kysymykset eivät johdattelisi vastaajia ja vastaukset olisivat mahdollisimman laaja-alaisia. Avoin vastausmahdollisuus tuo esille myös vastaajien tietämyksen ja tuntemuksen kysytystä aiheesta. Kappaleissa 3.3.1 – 3.3.2 on ensimmäisen kyselyn vastaukset kirjoitettu auki ja jaoteltu nykyisen menetelmän hyvät ja haasteelliset puolet. Haasteeksi koetut asiat on vielä jaettu projektin eri vaiheisiin. Kappaleessa 3.3.3 on kirjoitettu tarkemmin kvantitatiivisen-kyselyn tuloksista.

3.3.1 Hyvät puolet nykyisissä menetelmissä

Nykyisen projektinhallinnan hyvinä puolina pidettiin työn joustavuutta ja byrokratian helppoutta. Projektiorganisaation kevyt rakenne vähentää organisaation henkilöstökuluja. Tähän saakka johtamaton projektinhallinta on ollut kevyt, eikä sen omaksuminen ole vaatinut suurempaa paneutumista tai oppimista. Kevyt rakenne antaa periaatteessa mahdollisuuden reagoida nopeasti äkkitilanteiden aiheuttamiin muutoksiin.

3.3.2 Haasteet nykyisissä menetelmissä

Nykyisiin toimintatapoihin toivottiin kehitystä projektin eri vaiheissa. Myös selkeän projektipäällikön nimeäminen projekteihin katsottiin tarpeelliseksi. Seuraavissa kappaleissa käydään lävitse esille tulleet haasteet projektin eri vaiheissa.

3.3.2.1 Määrittelyvaihe

Määrittelyvaiheen haasteena on riittävän tarkan ja yksityiskohtaisen järjestelmäspeksin saaminen projektista. Alkuperäisen speksin määrittelee ideaalitapauksessa Metsysin edustaja yhdessä Roclan asiantuntijan ja loppuasiakkaan kanssa, mutta näin ei aina kuitenkaan ole ja speksit saattavat olla puutteellisia. Myös projektin tietojen tiedonkulku kaikille projektissa työskenteleville henkilöille on koettu haasteelliseksi. Kattavan työlistan puuttuminen projektilta vaikeuttaa projektin aikataulutusta ja hallintaa. Liian usean Metsysin henkilön osallistuminen määrittelyvaiheeseen saattaa aiheuttaa sekaannusta ja ristiin meneviä lupauksia.

3.3.2.2 Toteutusvaihe

Metsysin työskentely-ympäristö on haastava, johtuen useista päällekkäisistä projekteista. Koska projekteilla ei ole toteutusvaiheessa selkeää projektinhallintaa, on projektien seuraaminen ja kontrolloiminen vaikeaa. Tästä johtuen projektien tila karkaa helposti käsistä ja kukaan ei pysty sanomaan projektien valmiusastetta. Puutteellisesta seurannasta ja tiedonkulusta johtuen, aikataulut ja vastuualueet eivät aina ole projektihenkilöiden tai johdon tiedossa. Edellä mainituilla asioilla on varmasti vaikutus projektin tehokkuuteen ja kustannuksiin.

Riittävän projektidokumentoinnin tekeminen ja niiden päivittäminen projektin työvaiheista, työtehtävistä, toimintakaavioista ja työaikaseurannasta olisi tärkeää nykyisten ja tulevien projektien jatkuvan parantamisen kannalta.

Lisäksi toteutettujen projektien selkeä päättäminen, yhteenveto ja arvioinnit puuttuvat kokonaan.

3.3.2.3 Testaus

Testauksen ongelmana on ollut systemaattisen testauksen täydellinen puuttuminen. Ohjelmistonkehittäjä on itse testannut tekemänsä järjestelmän, mutta tämä testausmalli ei ole missään tapauksessa tarvittavan kattava ja aukoton. Järjestelmän puutteet ja viat on havaittu käyttöönoton yhteydessä tai käyttöönoton jälkeen järjestelmän ollessa jo tuotannossa. Vikojen ja puutteiden korjaaminen jälkikäteen on huomattavasti vaikeampaa, kuin järjestelmän kehitysvaiheessa.

Myös testausdokumenttien puute ja kiireellinen aikataulu ovat aiheuttaneet haasteita järjestelmän kattavaan testaukseen.

3.3.2.4 Ylläpito

Loppukäyttäjän käyttöjärjestelmän päivitykset ja muut muutokset saattavat aiheuttaa moninaisia ongelmia ohjelmistojen toimivuuteen. Takuu-aikana Metsysillä on sopimuksen mukaan velvollisuus korjata tulevat ongelmat. Loppukäyttäjän ”aiheuttamia” ongelmia ei Metsysin projektinhallinnalla tietenkään pystytä estämään, mutta kattava dokumentointi tehdystä ohjelmistosta auttaa mahdollisten ongelmien ratkaisemisessa. Kattavan dokumentoinnin puute tekee ylläpito-päivystämisestä myös haastavampaa.

3.3.3 Kvantitatiivisen-kyselyn tulokset

3.3.3.1 Ensimmäinen kysely ennen Scrum-projektia

Ennen Scrumin käyttöönottoa suoritetun kvantitatiivisen-kyselyn vastauksissa oli yksilöllistä hajontaa. Keskimäärin vastauksissa toivottiin kehitystä useaan eri osa-alueeseen projekteissa. Yleisesti nykyistä projektinhallintaa ei pidetty erityisen toimivana. Ongelmallisimpina pidettiin testauksen puutteellisuutta ja projektien läpikäymistä projektien päätyttyä. Projektien tarkempaan speksaamiseen, työlistaan ja sen selkeyteen toivottiin myös parannusta. Kaikki vastaajat uskoivat myös kokopäiväisen projektipäällikön nimeämisen selkeyttävän projektien läpivientä.

Melko tyytyväisiä oltiin edellisten projektien hyödyntämiseen alkavissa projekteissa ja mahdollisuuteen reagoida muuttuviin määrityksiin kesken projektien. Projektien valmiusaste oli melko hyvin tekijöillä selvillä. Metsysin projektit miellettiin melko läpinäkyviksi.

3.3.3.1 Toinen kysely projektin päätyttyä

Scrum-projektin loputtua suoritettiin kvantitatiivisen-kyselyn toinen osa. Samat monivalintakysymykset lähetettiin samoille henkilöille. Yksi vastaajista ei osallistunut Scrum-projektiin, joten hänen osaltaan toinen kysely jäi merkityksettömäksi. Tämän kyselyn vastauksissa pyydettiin painottamaan kokemuksia tehdystä Scrum-projektista.

Osa kysymyksistä ei liittynyt projektinhallintaan, joten näiden kysymysten osalta vastaukset noudattivat pitkälti ensimmäisen kyselyn vastauksia. Kysymyksiin jotka koskivat projektinhallintaa, saatiin muutoksia vastauksiin. Kaikki vastaajat olivat sitä mieltä, että Scrum-projektissa muutoksiin reagoiminen, tehtävien listaaminen, projektin läpinäkyvyys, projektin valmiusaste, töiden organisoiminen ja testaus olivat paremmin hoidettu, kuin aikaisemmissa projekteissa. Myös projektinhallintaa pidettiin selkeämpänä Scrum-projektissa, kuin aikaisemmissa projekteissa.

3.4 Scrumin soveltuminen Metsysin käytäntöihin

Kyselyjen ja omien havaintojen perusteella Scrum-menetelmä soveltuu teoriassa hyvin Metsysin työskentely-ympäristöön. Menetelmä tuo kaivattua läpinäkyvyyttä,

kommunikointia, töiden organisointia ja kattavan työlistan projekteihin. Lisäksi yrityksen johto ja muu henkilöstö pystyy halutessaan seuraamaan meneillään olevien projektien etenemistä.

Metsysin joustava työskentely-ympäristö pystytään säilyttämään myös Scrum-menetelmää käytettäessä. Scrum ei ota kantaa työskentelymetodeihin vaan metodit sovitaan Scrum-tiimin kesken, jokainen tiimin jäsen sitoutuu tämän jälkeen toimimaan yhdessä sovittujen käytäntöjen ja sääntöjen mukaan. Menetelmä ei myöskään ota kantaa miten käytännön ohjelmointityö toteutetaan vaan jokainen suorittaa valitut tehtävät parhaalla näkemällään tavalla. Tämä soveltuu hyvin ympäristöön missä työskentelee kokeneita ja ammattitaitoisia ohjelmistosuunnittelijoita.

Haasteita aiheuttaa usean projektin yhtäaikainen toteuttaminen ja ohjelmistosuunnittelijoiden sitominen samanaikaisesti useampaan projektiin. Toisaalta Scrum voi tuoda tähän selkeyttä ja mahdollisuuden reagoida nopeasti muuttuneisiin tilanteisiin. Lisäksi aikarajojen venyminen vanhoissa projekteissa aiheuttaa aikataulusongelmia uusissa projekteissa.

Metsysin projektit toteutetaan verrattain pienellä projektitiimillä, mukana on yleensä aktiivisesti vain 1-3 henkilöä, mutta hetkellisesti projektin parissa voi työskennellä 5-6 henkilöä. Scrum-menetelmä tukee parhaiten hieman isompia projektiryhmiä, tavanomaisesti Scrum-projektiryhmän koko on vähintään 5 henkilöä.

3.5 Scrum-koulutus

Allekirjoittanut opiskeli ensin itsenäisesti Scrum-menetelmää kirjoista ja Internetistä löytyvän tiedon avulla. Tämän jälkeen allekirjoittanut koulutti Metsysin henkilöstölle Scrumin toiminnan. Scrum-menetelmän opiskelu jatkui allekirjoittaneen ja muun henkilöstön osalta läpi pilottiprojektin ja toimintatavat hioutuivat yhä paremmin tukemaan Metsysin toimintamallia.

Henkilöstö suhtautui koulutukseen innostuneesti ja hyviä ideoita menetelmän toteuttamiseen nousi esiin koulutuksen yhteydessä. Avoimiksi ja epäselviksi jääneisiin kohtiin etsittiin vastaus alan kirjallisuudesta.

4 TOTEUTUS

Scrum pilottiprojektin aikana Metsysillä oli myös muita vastaavia projekteja työn alla ja nämä muut toteutettiin perinteisellä mallilla. Tämä toi lisä haastetta projektin aikataulutukseen ja resurssien saatavuuteen. Scrum-menetelmä on toki suunniteltu toimimaan ympäristössä missä on samanaikaisesti käynnissä useampia Scrum-projekteja, mutta toimiminen rinnakkain useamman eri hallintamallia käyttävän projektin kanssa oli lähtökohtaisesti haastavaa.

4.1 Wattensin projekti

Scrumin pilottiprojektiksi valikoitui sopivan kokoinen ja hyvään aikaan opinnäytetyön kannalta suoritettava Itävallan Wattensin paperitehtaan jälkikäsittelyn automatisointiprojekti. Tehdas valmistaa savukepaperia viidellä paperikoneella. Loppuasiakas haluaa automatisoida paperirullien ja tambuurikärryjen siirron paperikoneilta varastoihin, päällystysasemalle ja pituusleikkureille. Roclan asiantuntijat ovat laskeneet, että kyseiset tehtävät on optimaalista suorittaa kahdella AGV:llä (Automated Guided Vehicle). Tehtaalla on kahdeksan pituusleikkuria, neljä päällystysasema, yksi tambuurivarasto ja yksi automaattinen rullavarasto. Rullavarastona toimii ns. korkeapaikka-varasto jolla on oma tietojärjestelmänsä, liityntä MetRo-järjestelmästä varastoon oli määriteltävä ja koodattava erikseen. Järjestelmän käyttöönottoon oli varattu 15 päivää loppuasiakkaan luona.

4.2 Projektin aloitus

Projekti oli alkanut Metsysin osalta jo ennen Scrumin käyttöönottoa. Metsysin myyntipäällikkö oli käynyt tutustumassa loppuasiakkaan tehtaaseen ja kaupat oli jo solmittu. Rocla oli laatinut erittäin kattavan ja selkeän speksin alkavasta projektista ja palaveri Roclan kanssa joulukuussa 2010 selvensi hyvin vaatimuksia ja määrityksiä mitä projektiin liittyi. Järjestelmän käyttöönotto oli suunniteltu vuoden 2011

maaliskuun puoleen väliin. Korkeapaikkavarasto-linkin testaus sovittiin tammikuun loppuun.

Pidetyn palaverin, kirjattujen määritysten ja aikaisempien projektien pohjalta Scrum-tiimi laati ensimmäisen version tuotteen työlistasta. Työlista toteutettiin Excel-taulukolla ja siihen listattiin kaikki mahdolliset projektin työtehtävät, niiden kestot (aika-arvio) ja prioriteetit. Ensimmäisessä työlistan versiossa oli tehtäviä yhteensä 217 tuntia, tässä ei ollut otettu huomioon käyttöönottoon meneviä tunteja. Aikaa käyttöönottoon oli tällöin vajaa kolme kuukautta, joten aikataulu oli kohtalaisen tiukka.

Tuotteen työlistan ja muut Scrumin dokumentit päätimme toteuttaa Microsoft Excel –ohjelmalla. Excelillä pystyimme listaamaan työt ja laskemaan menetelmässä tarvittavat aika-arviot. Teimme kaikki neljä dokumenttia samaan Excel-dokumenttiin, ensimmäisellä sivulla oli tuotteen työlista ja edistymiskäyrä, jokainen sprintti ja sprintin edistymiskäyrä olivat omalla alisivulla.

Sovimme yrityksen johdon kanssa, että tälle projektille ja yleensäkin Metsysin projekteille soveltuu sprintin kestoksi kaksi viikkoa. Ennen käyttöönottoa ehditään toteuttamaan kuusi sprinttiä. Toinen sprintti oli poikkeuksellisesti vain puolitoista viikkoa, jotta saimme sprintit alkamaan ja loppumaan aina maanantaina.

Scrum-menetelmään kuuluu pitää päivittäin päiväpalaveri, jossa käydään nopeasti lävitse edistyminen edellisestä päivästä, koetut ongelmat ja työsuunnitelmat seuraavan päiväpalaveriin asti. Ajatuksena oli pitää päiväpalavereita säännöllisesti, mutta ei kuitenkaan päivittäin.

Emme järjestäneet erillistä Julkaisun suunnittelukokousta, koska kävimme kaikki projektiin liittyvät asiat lävitse Roclan kanssa aloituspalaverissa. Työlistaa laadittaessa projekti selkeni ja tehtävät tarkentuivat Scrum-tiimille.

Nimesimme yhden ohjelmistosuunnittelijan projektiomistajaksi, hän toimii myös kehitystiimin jäsenenä. Kehitystiimissä oli lisäksi kahdesta neljään muuta ohjelmistokehittäjää. Allekirjoittanut toimi pilottiprojektin Scrum-mestarina. Sovimme yhdessä yrityksen johdon, sekä Scrum-tiimin kanssa, että pilottiprojektissa

allekirjoittanut hoitaa myös osittain projektioimistajan tehtäviä. Katsoimme, että näillä lähtötiedoilla pystyimme aloittamaan projektin toteutuksen.

4.3 Projektin eteneminen

Projekti eteni verkkaisesti ensimmäiset kolme sprinttiä. Näiden sprinttien aikana saatiin suoritettua hieman päälle 40 tuntia projektintehtäviä. Neljännessä sprintistä lähtien töitä tehtiin nopeammalla tahdilla, mutta läpi projektin haasteita aiheutti ohjelmistokehittäjien siirtyminen projektista toiseen. Tämä aiheutti ongelmia saada tarvittavia resursseja toteuttamaan Wattensin projektia. Alun perin oli tarkoitus tehdä kuusi sprinttiä ennen käyttöönottoa, tähän ei erinäisistä syistä pystytty ja lopulta sprinttejä järjestettiin yksitoista.

Projektin edistymisen näki hyvin Scrum-menetelmän dokumenteista. Tuotteen – ja sprintin työlistasta pystyttiin havainnoimaan kuinka paljon ja mitä projektin töitä oli vielä tekemättä. Hankaluutta dokumenttien ylläpitoon toivat aika-arvioiden muutokset ja kesken projektin lisätyt tehtävät. Esim. Excelistä ulos saatu sprintin edistymiskäyrä vääristyi, jos tehtävään varattu aika-arvio alittui, koska kaaviossa verrattiin sprinttiin valittujen tehtävien kokonaisaika-arviota toteutuneisiin tunteihin. Tässä kappaleessa esitettävät sprintin edistymiskäyrät eivät siis kerro yksiselitteisesti paljonko sprintistä jäi töitä tekemättä vaan paljonko aika-arviosta jäätiin.

4.3.1 Sprintti 1 (5.1.2011–19.1.2011)

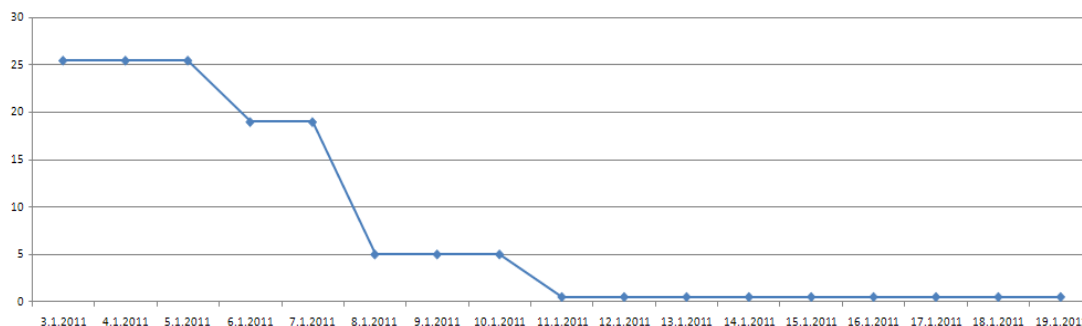
Ensimmäisessä Sprintin aloituspalaverissa valitsimme Sprintin työlistaon tarkoituksella vain vähän töitä, koska projektiin valitut suunnittelijat olivat vielä kiinni vanhoissa projekteissaan ja Wattensin aikataulu antoi tähän mahdollisuuden. Sprintin tehtäviksi valitsimme tietokantojen perustaminen ja liitynnän rakentamisen tehtaalla olevaan korkeapaikkavarastoon. Töitä sprinttiin valikoitui noin 30 tuntia. Kertasimme palaverissa myös Scrumin toimintamalleja ja suunnittelimme projektia kokonaisuudessa.

Sprintin aikana pidimme päiväpalavereita harvemmin kuin alun perin suunnittelimme, koska töitä eikä siten edistymistä tapahtunut tarpeeksi, että olisi ollut mielekästä ja järkevää pitää palavereita alkuperäisen suunnitelman mukaisesti.

Valittujen töiden eteneminen sujui jouhevasti, mutta pientä päänvaivaa aiheutti Sprintin työlistaan valittujen tehtävien aika-arvioiden pitäminen. Useat työvaiheet suoritettiin nopeammin, kuin mitä olimme arvioineet. Lisäksi osa valituista tehtävistä jatkui vielä tuleviin sprintteihin eli niitä emme pystyneet merkitsemään kokonaan tehdyiksi. Edellä mainitut asiat toivat lisä haastetta ylläpitää Scrumin dokumentteja, mutta Scrum-mestarin johdolla dokumentit saatiin pidettyä niin hyvin ajan tasalla, kun se oli mahdollista.

Sprintin aikana loppuasiakkaalta tuli lisäys speksiin. He halusivat liittynään MetRo-järjestelmästä heidän omaan MES (*Manufacturing Execution System*) – järjestelmään. Lisäsimme uudet työtehtävät tuotteen työlistaan ja arvioimme lisätöille aika-arviot, näitä tehtäviä emme aloittaneet tekemään ensimmäisessä sprintissä.

Ensimmäinen Sprintti päätettiin sovitus aikataulun mukaisesti. Emme järjestäneet Scrumiin liittyviä sprintin lopetuspalavereita, koska projektissa mukana olevat henkilöt olivat asiakkaiden luona hoitamassa edellisiä projektejaan. Sovimme kuitenkin, että sprintti oli päättynyt ja seuraava sprintti alkaa välittömästi edellisen päätyttyä. Ensimmäiseen sprinttiin valitut tehtävät saatiin suoritettua lähes täydellisesti. Osa yhdestä korkeapaikkavarastoon liittyvästä tehtävästä siirtyi tuleviin sprintteihin. Kuviossa 5 näkyy ensimmäisen sprintin toteutunut edistymiskäyrä.



KUVIO 5. Sprintti 1 edistymiskäyrä

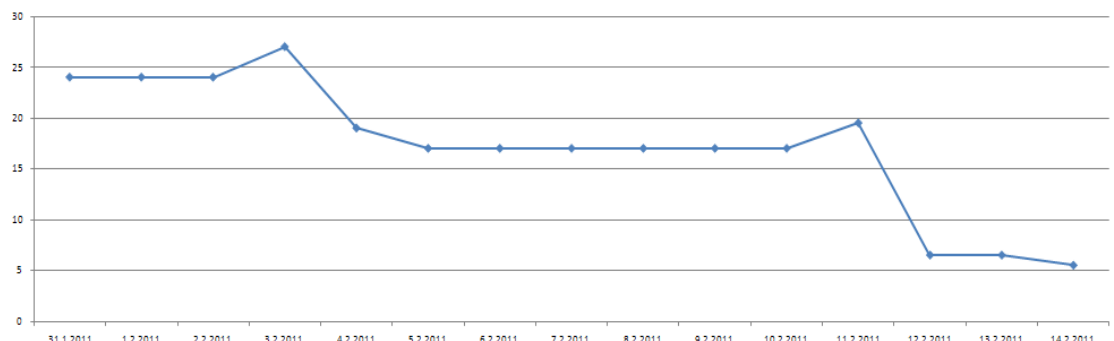
4.3.2 Sprintti 2 (19.1.2011 – 31.1.2011)

Toisen sprintin aloitusajankohtana tiimiläiset projektiomistajaa myöten olivat kiinni edellisissä projekteissaan ja näytti että he eivät ehdi tämän sprintin aikana tekemään töitä Wattensin projektiin. Sovimme Scrum tiimin kanssa, että tähän sprinttiin ei valita ainuttakaan tehtävää.

4.3.3 Sprintti 3 (31.1.2011–14.2.2011)

Projektiomistajan edellisen projektin käyttöönotto viivästyi vielä kahdella viikolla, joten myöskään tämän sprintin lähtökohdat eivät olleet parhaat mahdolliset. Myös muut kehitystiimin jäsenet olivat sprintin alussa kiinni muissa työtehtävissään. Sovimme sprintin aloituspalaverissa, että kehitystiimi toteuttaa tulevan kahden viikon aikana hieman yli 20 tuntia töitä tuotteen työlistasta.

Lisäsimme sprintin ensimmäisellä viikolla kaksi työtehtävää sprintin työlistaan. Tekijöiden resurssipulan vuoksi kehitystyötä päästiin tekemään vain osa-aikaisesti muiden töiden lomassa, välillä meni useampi päivä ilman edistystä Wattensin töissä. Sprintin loppuvaiheessa lisäsimme muutaman tehtävän työlistaan, jotta pääsimme tekemään tehtäviä eteenpäin. Sprintin tavoitteesta jäimme 6,5 tuntia, mutta lähtökohtaan nähden saimme projektia kohtalaisesti eteenpäin. Sprintti 3:n edistymiskäyrä näkyy kuviossa 6.



KUVIO 6. Sprintti 3 edistymiskäyrä

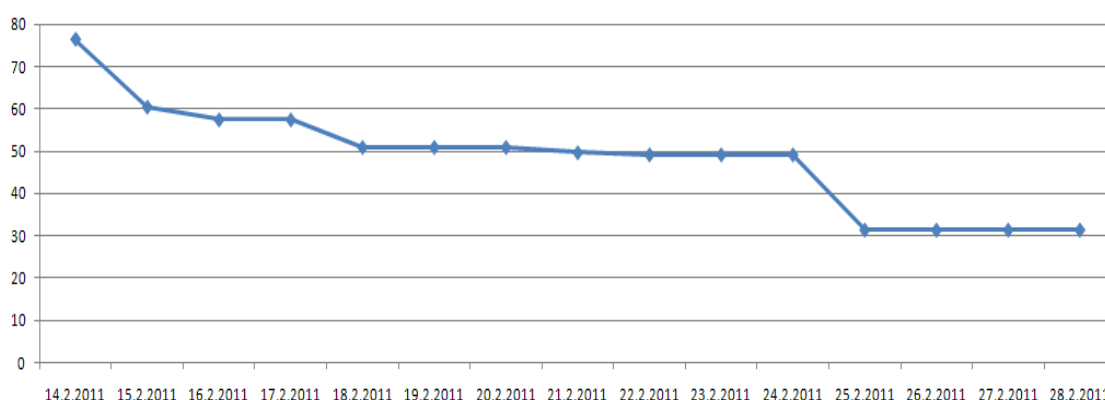
4.3.4 Sprintti 4 (14.2.2011 – 28.2.2011)

Neljännän sprintin alussa projektin valmiusaste oli 26 % ja tekemättömiä töitä oli jäljellä n. 180 tuntia. Sprinttiin valittiin tehtäviä yhteensä 76,5 tuntia. 4. Sprintin loppupuolelle oli sovittu asiakkaan kanssa FAT-testi. Testiin osallistuivat Metsysin -, Roclan - ja loppuasiakkaan edustajat, testissä testattiin lähinnä Roclan-järjestelmän toimivuutta, mutta myös Metsysin järjestelmältä vaadittiin toimivuutta testaamisen onnistumiseen.

Projektiomistajan edellisen projektin käyttöönotto venyi Metsysistä riippumattomista syistä ja projektiomistajan liittyminen Wattensin projektiin ei ollut mahdollinen. Teimme henkilöiden uudelleen organisointia Scrum-tiimissä. Alkuperäinen projektiomistaja jatkoi töitään vanhassa projektissaan ja kehitystiimistä nostettiin uusi projektiomistaja Wattensin projektiin. Myös kaikilla muilla Scrum-tiimin jäsenillä oli ”useampi rauta tulessa” läpi sprintin, tämä lisäsi haastetta sprintin tehtävien suorittamiseen. Sprintin tehtävien koodausta oli tekemässä kolme ohjelmistokehittäjää.

FAT-testin yhteydessä käytiin loppuasiakkaan kanssa tarkentavia keskusteluja tulevasta järjestelmästä. Keskusteluissa speksi tarkentui tietyiltä osin ja järjestelmään haluttiin muutamia muutoksia. Tarkennukset ja muutokset eivät vaikuttaneet merkittävästi projektin työkuormaan.

Saimme toteutettua sprintin töistä noin 58 %, tuleviin sprintteihin siirtyi hieman reilut 30 tuntia. Kuviossa 7 on nähtävissä 4. sprintin edistymiskäyrä. Töitä tehtiin melko epäsäännöllisesti läpi sprintin aikaisemmin mainittujen syiden takia. Alun perin sprinttiin valittiin hieman liian optimistisesti töitä olettaen, että tekijät pystyvät keskittymään pelkästään Wattensin projektiin.



KUVIO 7. Sprintti 4 edistymiskäyrä

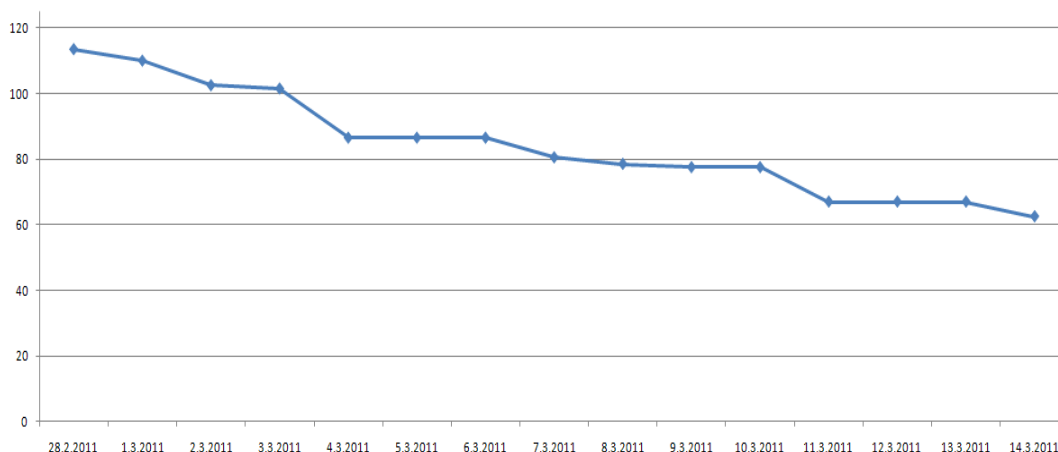
4.3.5 Sprintti 5 (28.2.2011 – 14.3.2011)

Projektin valmiusaste oli viidennen sprintin alussa 44.3 % ja työtä oli jäljellä arviolta 135 tuntia. Käyttöönoton alkuun oli kolme viikkoa aikaa ja aikataulu oli tiukka, mutta järjestelmällisellä toteutuksella mahdollinen. Kehitystiimin jäsenten talvilomat toivat lisää haastetta aikataulussa pysymiseen, resursseja toteutukseen oli edelleen niukasti.

Viidenteen sprinttiin valitsimme 113,5 tuntia töitä. Lähes puolet työmäärästä tuli kahdesta isosta tehtäväkokonaisuudesta (varastointi algoritmin teko) joita ei voinut jakaa pienempiin osiin. Roclan kanssa oli sovittu Metro-järjestelmän testaus 6. sprintin ensimmäiselle viikolla, joten ohjelmisto piti saada 5. sprintin aikana testausvalmiiksi.

Sprintin aikana loppuasiakas halusi joitakin muutoksia ja lisäyksiä järjestelmään, tämä toi lisätöitä arviolta 10 tunnin verran. Lisäsimme uudet tehtävät tuotteen työlistaan ja siirsimme osan uusista tehtävistä kuluvan sprintin työlistaan.

Sprintin suuresta työmäärästä saimme toteutettu alle puolet (44,30 %) ja toteuttamatta jäi 66 tuntia. Suuri osa tekemättä jääneistä töistä koostui edellä mainituista kahdesta isosta tehtäväkokonaisuudesta ja käyttöliittymien muokkaamisesta. Tärkeimmät toiminnallisuudet tulevia testejä varten saatiin kuitenkin valmiiksi. Sprintin edistymiskäyrä näkyy kuviossa 8.



KUVIO 8. Sprintti 5 edistymiskäyrä

4.3.6 Sprintti 6 (14.3.2011 – 28.3.2011)

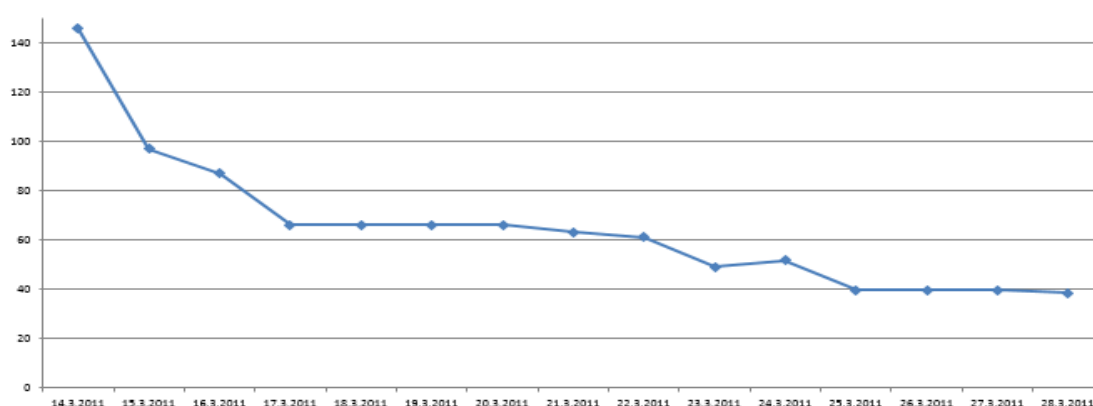
Alkuperäisen aikataulusuunnitelman mukaan käyttöönotto olisi pitänyt olla tämän sprintin aikana, mutta järjestelmän käyttöönotto siirtyi eteenpäin Metsysistä johtumattomista syistä. Koko projektin valmiusaste ennen sprinttiä oli noin 61 % ja työtehtäviä oli arviolta jäljellä 75 tuntia. Järjestelmä oli pääpiirteissään testauskunnossa, mutta ohjelmiston hienosäätöä ja dokumentointia oli vielä paljon jäljellä. Sprinttiin valittiin kaikki tuotteen työjonon tehtävät. Sprintin aikana tehtäviä lisättiin työlistalle lähes tuplasti. Lisä tehtävät koostuivat pääosin FAT-testistä, mitä ei ollut otettu huomioon tuotteen työlistaa tehdessä. Sprintin työlistan tehtävämäärä nousi lopulta 146,5 tuntiin. Töitä ennen käyttöönottoa oli vielä paljon, mutta usko tehtävistä suoriutumiseen oli vahva.

Sprintin toisena päivänä Roclan asiantuntijat tulivat testaamaan järjestelmää, Metsysiltä testaukseen osallistui Scrum-tiimi. Ohjelmisto toimi hyvin ja esille tulleet ongelmat saatiin korjattua testauksen yhteydessä. Rocla halusi lisäksi ohjelmistoon muutamia muutoksia ja lisäyksiä. Suurin osa toiveista saatiin lisättyä ohjelmistoon testien aikana, loput suoritettiin testin jälkeen. Testi kokonaisuudessaan oli työtä, jota ei ollut otettu huomioon työlistaa tehdessä. Testi ja muutokset lisäsivät töitä yhteensä 44,5 tunnin verran. Roclan kanssa sovittiin, että seuraava testipäivä järjestetään juuri ennen järjestelmän käyttöönottoa.

Loppuasiakkaan pyynnöstä päällystysasemat otetaan käyttöön vasta syksyllä 2011, tämä vähensi hieman meidän työkuormaa keväältä. Lisäksi loppuasiakkaalta tuli muutospyyntöjä järjestelmään. Heidän ehdottamalla tavalla toteutettuna muutokset

olisivat tuoneet projektin tässä vaiheessa kohtuuttoman suuren muutostyön Metsysille. Projektioimistaja, Roclan edustaja ja loppuasiakas neuvottelivat muutoksista ja lopulta päädyttiin luopumaan osasta halutuista muutoksista. Järjestelmän muutoksista aiheutuneet lisätehtävät lisättiin päiväpalaverissa tuotteen työlistalle ja sieltä sprintin työlistalle. Muutokset toivat lisätoita lopulta vain 6 tuntia.

Sprintin tehtävistä saatiin tehtyä yli 70 % ja järjestelmä oli käyttöönotto kunnossa. Tekemättömät tehtävät oli tarkoitus tehdä käyttöönoton yhteydessä. Kuviosta 9 selviää kuudennen sprintin edistyminen.



KUVIO 9. Sprintti 6 edistymiskäyrä

4.3.7 Sprintti 7 (28.3.2011 – 11.4.2011)

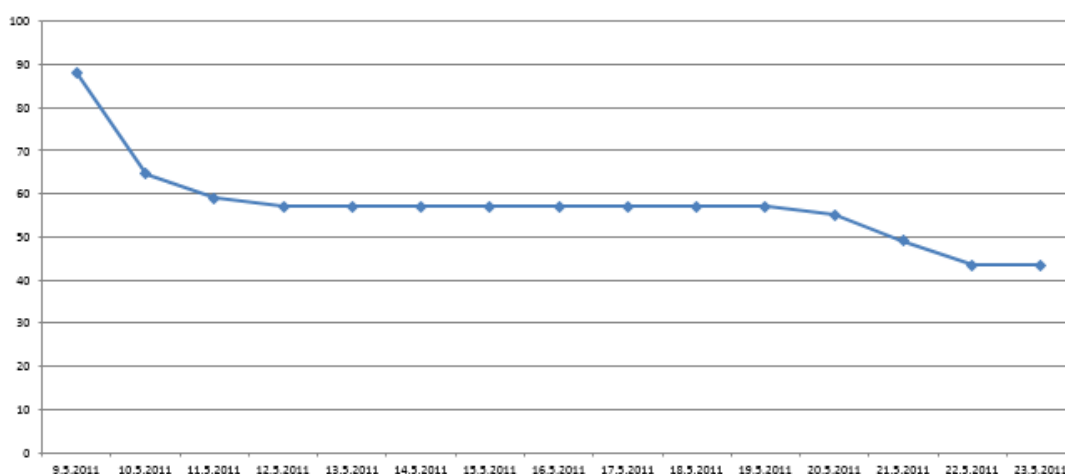
Projektin valmiusaste oli sprintin alussa noin 71 %. Jäljellä olevista tehtävistä suuri osa koostui dokumenttien teosta ja muutamasta työstä, mitkä pystyttiin hoitamaan vain käyttöönoton yhteydessä. Valmiusastetta vääristi kaksi isoa tehtäväkokonaisuutta joihin oli varattu paljon tunteja (20 h ja 13,5 h), tehtäviä ei voinut merkitä valmiiksi ennen järjestelmän käyttöönotto-testausta. Tehtäviä 7. sprinttiin valittiin 88 tuntia. Tämän sprintin aikana suoritettiin järjestelmän käyttöönotto Itävallassa loppuasiakkaan paperitehtaalla.

Sprintin ensimmäisenä päivänä järjestelmää testattiin vielä kertaalleen Roclan kanssa. Testaamiseen ja testaamisen aikana esille tulleiden puutteiden ja virheiden korjaamiseen kului yhteensä 22 tuntia. Testi ja lisätehtävät lisättiin tuotteen työlistaan ja 7. sprintin työlistaan.

Yksi Scrum-tiimin jäsenistä lähti loppuasiakkaan tehtaalle heti sprintin ensimmäisen viikon alkupuolella ja aloitti järjestelmän käyttöönoton. Muut projektin asiantuntijat avustivat käyttöönottoa etäyhteyden avulla. Käyttöönottoon oli varattu 15 työpäivää. Käyttöönoton ensimmäisellä viikolla oli tarkoitus asentaa MetRo-järjestelmä tehdasympäristöön ja suorittaa testiä todellisessa ympäristössä. Jälkimmäisellä viikolla oli tarkoitus ottaa järjestelmä tuotantokäyttöön ja kouluttaa käyttäjät käyttämään Metro-järjestelmää.

Asiat eivät kuitenkaan menneet, kuten oli suunniteltu ja käyttöönotossa ilmaantui ongelmia sekä MetRo-järjestelmässä, että loppuasiakkaan järjestelmissä. Lisäksi loppuasiakkaan jatkuvasti käynnissä oleva tuotanto hankaloitti uuden järjestelmän järjestelmällistä käyttöönottoa ja testaamista. Edellä mainitut ongelmat aiheuttivat sen, että MetRo-järjestelmää ei saatu käyttöönotettua suunnitellussa 15 päivässä vaan järjestelmän käyttöönottoa täytyi jatkaa vielä uudella vierailulla loppuasiakkaan tehtaalla. Toinen käyttöönotto oli tarkoitus järjestää noin kuukauden kuluttua ensimmäisestä käyttöönotosta.

Sprintin tehtävistä pystyttiin tekemään vain noin 50 %. Tekemättömien tehtävien suorittamista jatkettiin seuraavissa sprinteissä ja seuraavassa käyttöönotossa. Sprintin 7. edistymiskäyrä näkyy kuviossa 10.



KUVIO 10. Sprintti 7 edistymiskäyrä.

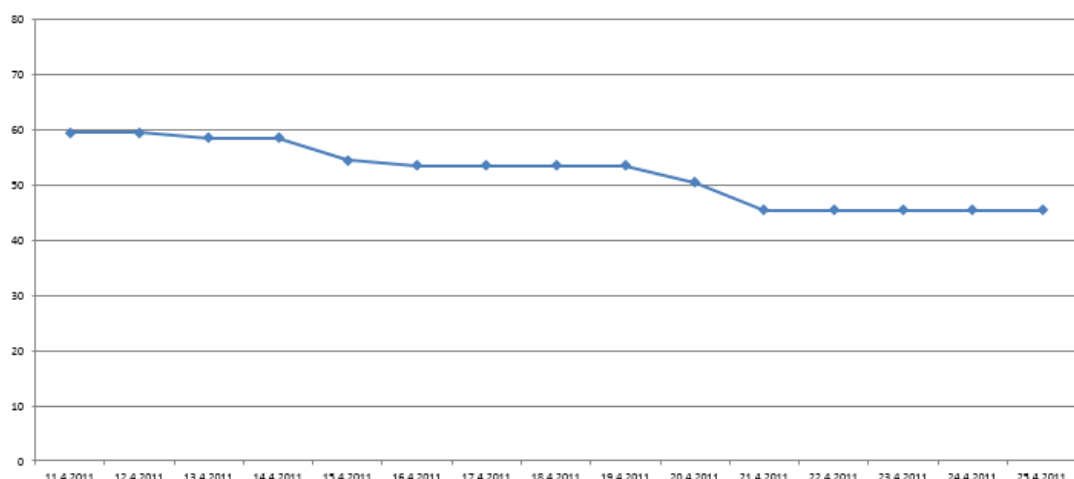
4.3.8 Sprintti 8 (11.4.2011 – 25.4.2011)

Kahdeksannen sprintin alussa projektin valmiusaste oli tippunut 62 % johtuen projektiin tulleista lisätehtävistä. Lisätehtävä liittyivät sekä järjestelmä muutoksiin, että dokumenttien tekoon, mitkä loppuasiakas vaatii toimitettavaksi järjestelmän mukana. Sprintin aloituspalaverissa tähän sprinttiin valittiin tehtäviä 59,5 tuntia.

Sprintin ensimmäisellä viikolla Scrum-tiimin jäsen oli suorittamassa ensimmäistä käyttöönottoa loppuasiakkaan luona. Sprintin jälkimmäisellä viikolla testattiin käyttöönotossa havaittuja virhetilanteita simulaattorin avulla ja viimeisteltiin järjestelmädokumentteja loppuasiakkaalle. Käyttöönotossa mukana ollut Scrum-tiimin jäsen suoritti edellä mainittuja töitä. Myös tähän sprinttiin valittiin kaksi isoa tehtäväkokonaisuutta, mutta niitä ei päästy tekemään sprintin aikana, suurelta osin näiden tehtävien takia sprintissä jäi suorittamattomia työtunteja verrattain paljon.

Muut tiimin jäsenet olivat siirtyneet tekemään pääsääntöisesti uusia projekteja, mutta tarvittaessa auttoivat myös tässä projektissa. Scrumiin liittyviä palavereita ei ollut tarvetta tässä sprintissä järjestää niin usein, kuin projektin aikaisemmissa sprinteissä.

Sprintin tehtävistä saatiin suoritettua vain reilu 20 % ja töitä jäi tuleviin sprintteihin noin 40 tuntia. Edistymiskäyrä on nähtävissä kuviossa 11.



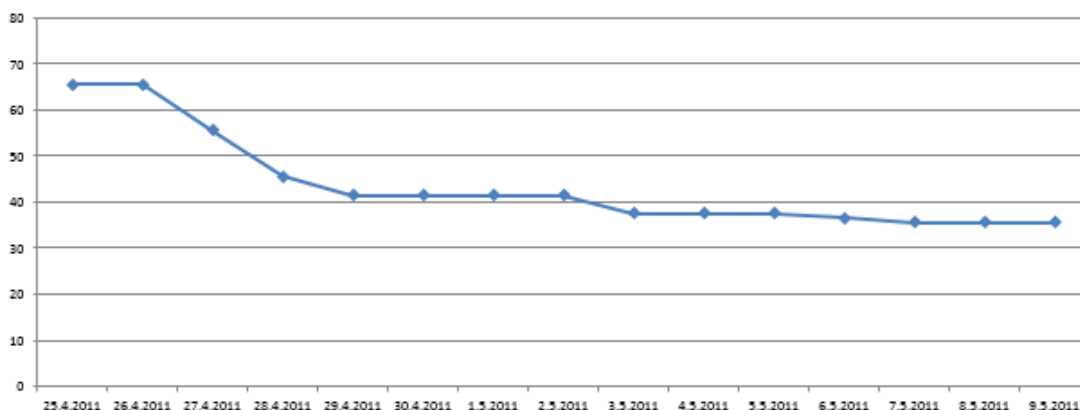
KUVIO 11. Sprintti 8 edistymiskäyrä.

4.3.9 Sprintti 9 (25.4.2011 – 9.5.2011)

Yhdeksäs sprintti sijoittui ensimmäisen ja toisen käyttöönoton väliin, sprinttiin valittiin tehtäviä 65,5 tuntia. Projektin valmiusaste ennen sprinttiä oli 64 %. Vaikka valmiusaste oli verrattain matala, oli järjestelmä kuitenkin toimintakunnossa ja jäljellä olevat tehtävät liittyivät dokumenttien tekoon ja kahteen isoon tehtäväkokonaisuuteen, jotka voitiin suorittaa vain seuraavan käyttöönoton yhteydessä. Sprintin tehtävät koostuivat ensimmäisessä käyttöönotossa havaittujen virheiden ja ongelmien korjaamisesta ja testaamisesta. Loppuasiakkaalta tuli myös joitakin muutospyyntöjä järjestelmään.

Käyttöönotossa mukana ollut Scrum-tiimin jäsen viimeisteli loppukäyttäjäohjetta kielikäännöstä varten. Tehtäviä ei ollut määrällisesti paljon ja myös tämän sprintin aikana suuri osa Scrum-tiimin jäsenistä työskenteli muiden projektien parissa, tarvittaessa kuitenkin auttaen myös Wattensin projektin tehtävissä.

Sprinttiin valituista tehtävistä saatiin suoritettua 30 tuntia mikä oli 46 % sprintin tehtävistä. Edistymiskäyrä näkyy kuviossa 12.



KUVIO 12. Sprintti 9 edistymiskäyrä.

4.3.10 Sprintti 10 (9.5.2011 – 23.5.2011)

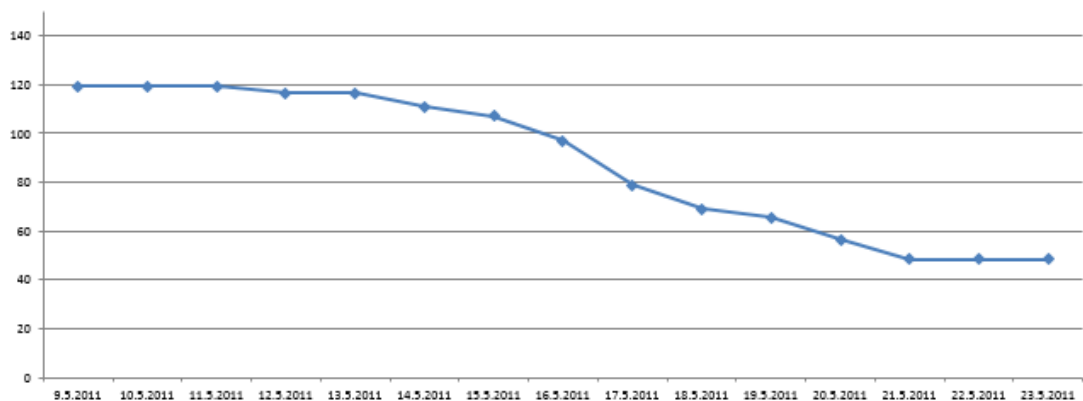
Kymmenennen sprintin aikana järjestettiin järjestelmän toinen käyttöönotto. Käyttöönottoon oli varattu kaksi työviikkoa aikaa. Sprintille valittiin aluksi tehtäviä noin 50 tuntia, mutta sprintin edetessä tehtäviä lisättiin työjonoon, siten että lopulta

sprintin työlistassa oli tehtäviä yhteensä 116,5 tuntia. Sprintin alussa projektin valmiusaste oli hieman yli 70 %.

Sprintin kuluessa korjattiin järjestelmässä olleita puutteita ja lisättiin asiakkaalta tulleita muutoksia järjestelmään. Tehtävät lisättiin sprintin työlistalle sitä mukaan, kun ne havaittiin ja niitä suoritettiin Metsysin konttorilla muiden Scrum-tiimin jäsenten toimesta. Tässä vaiheessa Scrumiin kuuluva selkeä tehtävien listaus auttoi osaltaan töistä suoriutumiseen.

Sprintin aikana järjestelmä saatiin saatettua toimivaan kuntoon ja loppuasiakas pystyi käyttämään järjestelmää itsenäisesti tuotantokäytössä. Järjestelmässä oli edelleen joitakin ongelmia mm. kommunikoinnissa korkeapaikkavaraston kanssa, mutta pääsääntöisesti järjestelmä toimi. Käyttöönoton aikana oli tarkoitus myös kouluttaa Metro-järjestelmää loppuasiakkaan operaattoreille ja pääkäyttäjille, mutta kiireellisen aikataulun takia koulutusta ei ehditty järjestämään. Koulutusta varten täytyi tehdä erillinen koulutusmatka loppuasiakkaan luokse.

Kuviosta 13 näkyy sprintin edistyminen, sprintin työlistan 116,5 tunnista jäi tekemättä 26,5 tuntia. Sprintin aikana tehtiin pitkiä työpäiviä, jotta kaikki tarvittavat muutokset saatiin valmiiksi sovituissa aikarajoissa. Projekti eteni hyvin tämän sprintin aikana.



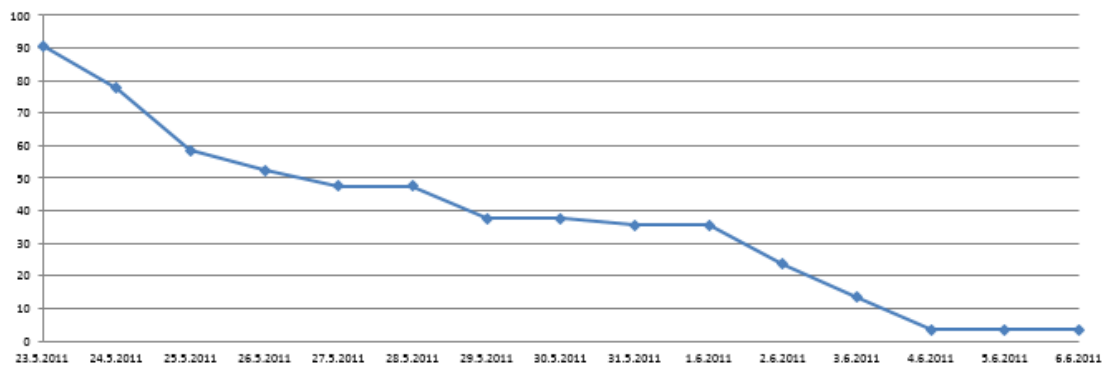
KUVIO 13. Sprintti 10 edistymiskäyrä.

4.3.11 Sprintti 11 (24.5.2011 – 23.5.2011)

10. Sprintin jälkeen koko projektin valmiusaste oli 78 %. 11. Sprinttiin valittiin tehtäviä kaikkiaan 90,5 tuntia. Järjestelmä oli ollut jatkuvassa käytössä loppuasiakkaan tehtaalla eli puuttuvat tehtävät eivät olleet merkittäviä järjestelmän päivittäisen toiminnan kannalta. Sprinttiin valituista tehtävistä suuren osan muodostivat toisessa käyttöönotossa havaitut virheet ja loppuasiakkaan vaatimat muutokset, nämä tehtävät lisättiin ensin projektin työlistaan ja sieltä sprintin työlistalle.

Sprintin aloituspalaverissa kävimme lävitse tuotteen työlistan ja lisäsimme kaikki työt tämän sprintin työlistalle, tältä listalta jaoimme tehtävät Scrum-teamin jäsenten kesken. Osa Scrum-tiimin jäsenistä teki muutoksia ohjelmaan ja osa työsti vaadittavia dokumentteja valmiiksi. Järjestelmäpäivitykset hoidettiin etäyhteyden avulla, koska paikanpäällä loppuasiakkaan luona ei ollut Metsysin asiantuntijoita.

Sprintin tehtävistä suoritettiin 96 % ja jäljelle jäi vain 3,5 tuntia aika-arviosta. Sprintti 11 jäi viimeiseksi tämän projektin osalta, jäljelle jääneitä tehtäviä oli sen verran vähän, että niiden takia ei ollut mielekästä ylläpitää Scrum-projektia. Sprintin 11 edistymiskäyrä näkyy kuviossa 14.



KUVIO 14. Sprintti 11 edistymiskäyrä.

4.4 Projektin päättäminen

Projekti saatiin lopulta Scrumin osalta päätökseen 11. sprintin jälkeen. Projekti oli venynyt erinäisistä syistä suunnitellusta 6 sprintin projektista 11 sprintin mittaiseksi.

Järjestelmäkoulutukset sovittiin pidettäväksi pian toisen käyttöönottomatkan jälkeen. Järjestelmään speksatun päällystysasemien ohjauksen toteuttaminen sovittiin tehtäväksi seuraavana syksynä. Näitä töitä varten oli turha jättää projektia auki vaan ne päätettiin toteuttaa erillisinä töinä sitten, kun asemien käyttöönotto tulee ajankohtaiseksi.

5 TULOKSET

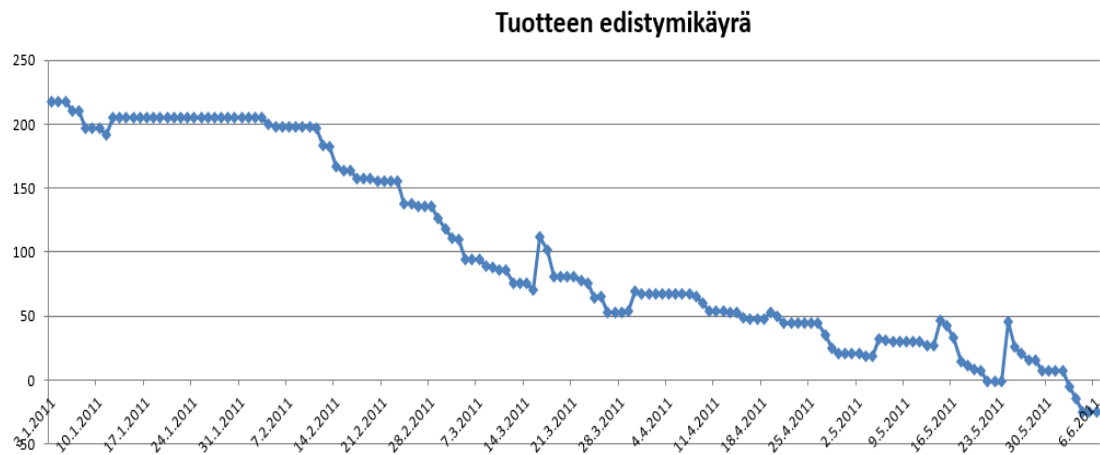
Saimme toimitettua toimivan automaattitruckien ohjausjärjestelmän loppuasiakkaalle, joten tältä osin projekti oli onnistunut. Aikataulu järjestelmän toimittamiseen venyi muutamalla kuukaudella, mutta tämä ei johtunut ainoastaan Metsysin järjestelmän toimitusvaikeuksista, vaan ongelmia oli kaikilla projektin osapuolilla. Uskon, että tarvittaessa Metsysin puolesta projekti olisi saatu valmiiksi myös alkuperäisessä aikataulussa, toki tämä olisi vaatinut vielä enemmän ponnisteluja tiimiltä.

Tekemiemme käyttöönottojen osalta olisimme voineet suoriutua paremmin. Kuten aikaisemmin on mainittu, jouduimme tekemään kaksi käyttöönottoa suunnitellun yhden sijasta. Käyttöönottojen välissä olisi pitänyt saada järjestelmä saatettua valmiimpaan vaiheeseen, nyt jouduimme toisen käyttöönoton aikana tekemään suuria ponnisteluja, jotta järjestelmä saatiin luovutettua loppuasiakkaalle.

Osaltaan haasteita aiheuttivat loppuasiakkaan jatkuvat muunnospyynnöt järjestelmään kesken projektin. Muutoksiin pitää pystyä reagoimaan hallitusti, mutta myöskään kaikkia muutoksia ei tarvitse hyväksyä. Se, mitä muutoksia lähdetään toteuttamaan ja mitkä jätetään toteuttamatta, on aina erittäin hankala kysymys. Tähän kun vielä lisätään kysymykset kuka muutokset maksaa, ollaan hankalan asian äärellä. Toisaalta Scrum-menetelmähän on tarkoitettu ketteräksi projektinhallinnaksi, jossa on mahdollista reagoida nopeasti muuttuneisiin projektin olosuhteisiin, joten näistä muunnospyyntöistä pitäisi pystyä suoriutumaan, jos niitä aloitetaan tekemään.

Koko projektin edistymistä pystyy helpoiten tarkastelemaan tuotteen kehityskäyrästä. Kehityskäyrä on kuvattu kuviossa 15. Alun perin projektin tehtävien arvioitu

kokonaisaika oli 217 tuntia, lopulta koko projektiin tehtiin Scrum-dokumentaation mukaan töitä 386,5 tuntia eli 78 % enemmän kuin alun perin arvioitiin.



KUVIO 15. Julkaisun edistymiskäyrä.

Projekti eteni ensimmäiset kolme sprinttiä rauhallisesti johtuen tiimiläisten kiireistä muissa projekteissa. 4. Sprintin loppupuolelle tuli loppuasiakkaan edustajia suomeen ja tällöin Metron osalta piti olla kaikki käyttöliittymät ja oleelliset toiminnallisuudet valmiina esiteltäviksi.

Roclan asiantuntijat tulivat Metsysin toimistolle maaliskuun puolessa välissä testaamaan järjestelmää. Itse testiin kuluva aika ei ollut otettu huomioon projektin aika-arviossa, tästä johtuen 14.3. on 41 tunnin lisätyö edistymiskäyrällä. Testien yhteydessä lisättiin myös muutamia tehtäviä tuotteen työlistalle.

Ensimmäinen käyttöönotto suoritettiin maaliskuun lopulla sprintin 7 aikana, sen hetkisen käsityksen mukaan järjestelmä oli käyttöönottokunnossa. Kuten yllä on kirjoitettu, ei järjestelmää saatu käyttöönotettua tällä käyttöönottomatkalla vaan jouduttiin tekemään vielä toinen käyttöönotto toukokuun alussa, sprintin 10 aikana.

Viimeisten kolmen sprintin aikana projektiin lisättiin tehtäviä hieman reilu 100 tuntia, tässä oli mukana korjauksia järjestelmään sekä uusia ominaisuuksia loppuasiakkaan pyynnöstä. 100 tunnin lisätöistä suoriutuminen alkuperäisten töiden lisäksi vaati isoja ponnisteluja Scrum-tiimiltä ja lopulta töihin käytettiin 24,75 tuntia enemmän aikaa, kun oli arvioitu. Tästä johtuen tuotteen edistymiskäyrä painuu negatiiviseksi projektin lopussa.

Lopulta 11 sprintin jälkeen järjestelmä oli käyttöönotettu asiakkaan tehtaalla Wattensissa. Seuraavalle syksylle jäi vielä päällystysasemien käyttöönotto, mutta tähän liittyvä työ oli jo osittain tehty järjestelmään.

Käyttöönotettu järjestelmä poikkesi melko paljon alun perin speksatusta järjestelmästä, mutta näin se yleensä menee. Projektin alussa, kun järjestelmiä määritellään, ei vielä tiedetä, eikä osata ottaa huomioon kaikkia yksityiskohtia mitä toimiva järjestelmä vaatii. Tämä on tavallaan ihan ymmärrettävää, mutta aiheuttaa lisä haasteita meille järjestelmän toimittamiseen. Juuri näiden projektien muuttuvien määreiden takia Scrum-menetelmän tai jonkun vastaavan menetelmän käyttö projektinhallinnassa on perusteltua.

6 TULOSTEN ANALYSOINTI

Tuloksia analysoitaessa pyrin analysoimaan projektia Scrumin näkökulmasta. Analysoin erikseen Scrumin tuomat edut, Scrumin aiheuttamat haasteet, vaikutukset tuotteen laatuun ja lyhyesti myös vaikutukset projektin kustannuksiin.

6.1 *Scrum-menetelmän edut*

Scrumin edut Metsysin projektinhallinnassa olivat monipuolisia. Yksi näkyvimmistä muutoksista perinteisiin projekteihin verrattuna oli lähes päivittäiset, nopeat 15 minuutin päiväpalaverit projektiryhmän kesken. Toinen näkyvä muutos aikaisempiin projekteihin oli menetelmään kuuluvat työlistaukset, joiden avulla projektin etenemistä oli vaivatonta seurata.

Lisähyötynä Metsys sai projektin töistä suuntaa-antavat tunti-arviot ja toteutuneet tunnit, jotka kirjattiin Scrumin dokumentaatioon. Aikaisemmissa projekteissa Metsys ei ole pitänyt tarkkaa tuntiseurantaa.

6.1.1 Sprintin suunnittelukokous

Sprintin suunnittelukokouksessa oli mukana yleensä koko scrum-tiimi, palaveri pidettiin maanantai aamupäivisin sprintin ensimmäisenä päivänä. Poiketen Scrumin ohjeistuksesta, pidimme suunnittelukokouksen yhdessä osassa, jossa valitsimme tehtävät tulevaan sprinttiin ja myös ideoimme, kuinka tehtävät toteutetaan. Suunnittelukokoukset kestivät noin 2 tuntia.

Suunnittelukokouksen jälkeen kaikilla tiimin jäsenillä oli selkeänä tiedossa mitä kukin tiimiläinen alkaa tekemään ja kaikki olivat ajan tasalla projektista. Scrum-mestarina toiminut allekirjoittanut muodosti valituista tehtävistä sprintin työlistan ja asetti jonon jakoon tietoverkkoon kaikille tiimin jäsenille.

6.1.2 Päiväpalaveri

Sovimme projektin alussa Scrum-tiimin kanssa, että emme pidä päivittäin päiväpalaveria vaan aina tarpeen mukaan. Käytännössä palaverit pidettiin kolme kertaa viikossa: maanantaisin, keskiviikkoisin ja perjantaisin. Sprintin aloitusviikolla maanantain palaveri pidettiin tiistaisin.

Päiväpalaverissa on tarkoituksena käydä asiat 15 minuutissa lävitse kolmen kysymyksen avulla joihin kaikki Scrum-tiimin jäsenet vastaavat. Tämän käytännön avulla jokainen tiimin jäsen pysyi ajan tasalla mitä kukin tekee ja jos jollain oli ongelmia töissä etenemisessä pystyivät muut jäsenet auttamaan palaverin jälkeen. Tämä osaltaan nopeutti projektin etenemistä, koska kukaan ei jäänyt itse painimaan ongelmien kanssa vaan niitä ratkottiin yhdessä, jollain tiimiläisellä saattoi olla kokemuksia samoista ongelmista ja näin ollen myös ratkaisu ongelmiin.

Jos jotain akuutteja lisätoita tuli projektiin, lisättiin ne sprintin työlistalle näissä palaverissa. Näitä tilanteita oli varsinkin käyttöönottojen yhteydessä, jolloin ei ollut mahdollisuutta odottaa seuraavaan sprinttiin töiden aloittamista.

Scrum-mestari päivitti sprintin työlistan päiväpalaverissa tulleiden tietojen avulla, toki työlistaa päivitettiin myös muulloin, mutta pääsääntöisesti aina tämän palaverin jälkeen.

Päiväpalaveri oli myös oiva tilaisuus Scrum-tiimin ulkopuolisille henkilöille kuulla projektin edistymisestä. Näihin palaveriinkin oli kaikilla lupa tulla kuuntelemaan, mutta vain tiimiläiset saivat käyttää puheenvuoroja.

6.1.3 Sprintin katselmointi ja – jälkitarkastelu

Yhdessä sovitulla päätöksellä pidimme Scrum-menetelmään kuuluvat Sprintin katselmoinnin ja Sprintin jälkitarkastelun yhdessä palaverissa aina sprintin viimeisenä perjantaina. Tässä palaverissa kävimme lävitse tehtävät sprintin työlistasta ja keskustelimme mitä olimme saaneet aikaseksi ja mitä tehtäviä sprintistä jäi toteuttamatta ja miksi. Näiden palaverien suurin anti oli sprintin analysointi ja siitä oppiminen tuleviin sprintteihin. Näihin palaveriinkin käytimme aikaa yleensä kahdesta kolmeen tuntiin, valitettavasti kyseistä palaveria ei erinäisistä syistä pystytty järjestämään jokaisen sprintin jälkeen.

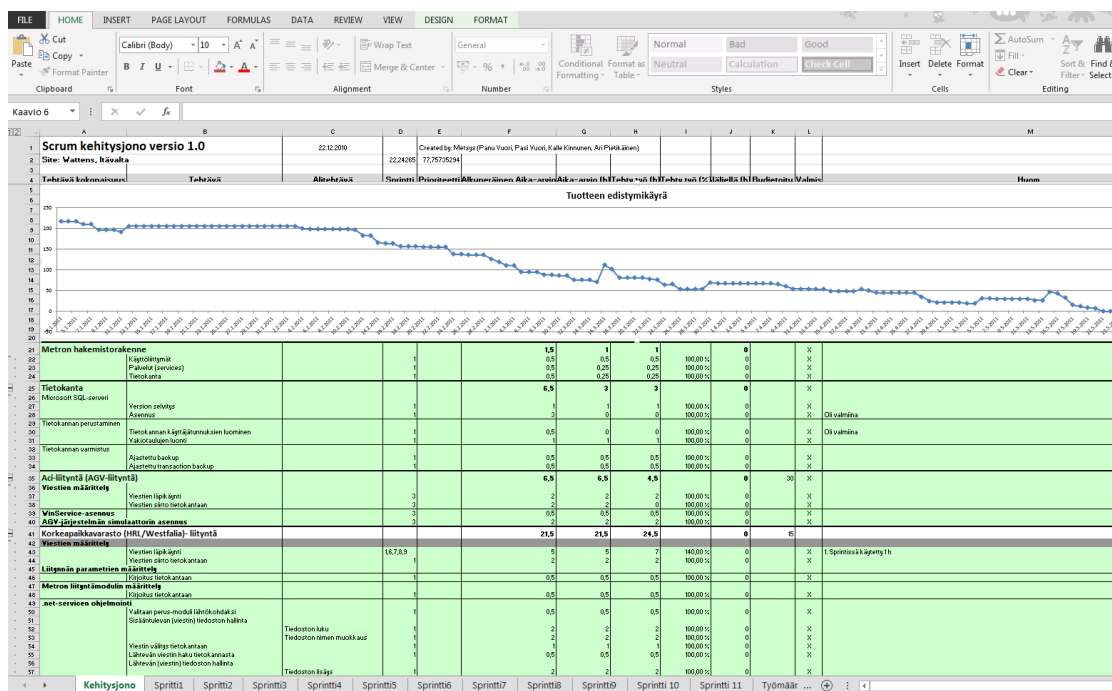
Sprintin jälkitarkastelu palaveri olisi ollut hyvä järjestää jokaisen sprintin jälkeen. Palaverissahan analysoidaan tiimin kanssa suoritettujen sprinttien töitä ja työskentelytapoja. Näiden palaverien ansiosta saataisiin jatkuvaa kehitystä projektin töiden tekemiseen ja työskentelytapoihin.

6.1.4 Scrumin dokumentit

Scrumiin kuuluu neljä eri dokumenttia mitkä liittyvät kaikki toisiinsa. Dokumentit ovat: tuotteen työlista, julkaisun edistymiskäyrä, sprintin työlista ja sprintin edistymiskäyrä. Koska tämä oli Metsysin pilottiprojekti Scrum-menetelmällä, piti meidän suunnitella ja toteuttaa kyseiset dokumentit tätä projektia varten. Suunnittelimme dokumentit yhdessä Scrum-tiimin jäsenten kanssa ja toteutuksen teimme Microsoftin Excel-ohjelmalla. Kaikki neljä dokumenttia oli yhdessä Excel-dokumentissa, jokainen sprintti oli eroteltu omille välilehdille.

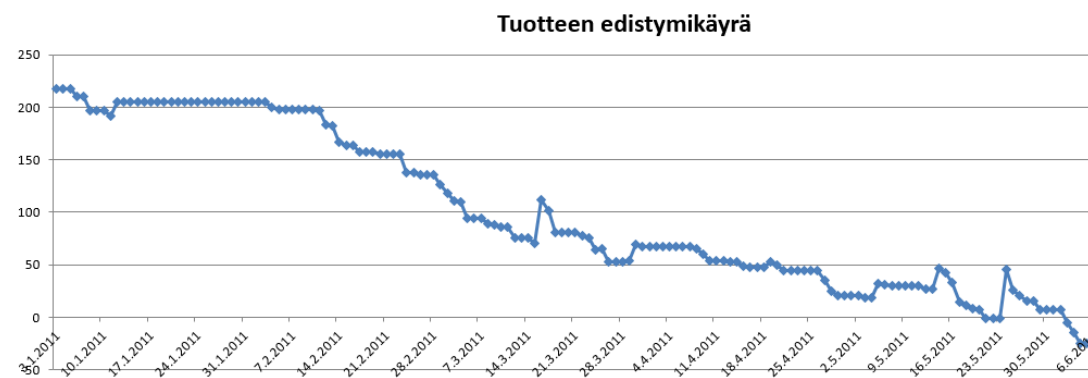
Tuotteen työlistaan listasimme projektin alussa kaikki tehtävät mitä sillä hetkellä tiesimme projektiin kuuluvan. Projektin edistyessä päivitimme työlistan tehtävien statuksen ja tarvittaessa lisäsimme tehtäviä siihen. Työlistasta oli kaikkien halukkaiden helppo seurata mitkä tehtävät oli tehty, mitkä olivat työn alla ja mitä ei

ollut vielä aloitettu. Kuviossa 16 on esitetty meidän käyttämä tuotteen työlista ja julkaisun edistymiskäyrä.



KUVIO 16. Tuotteen työlista.

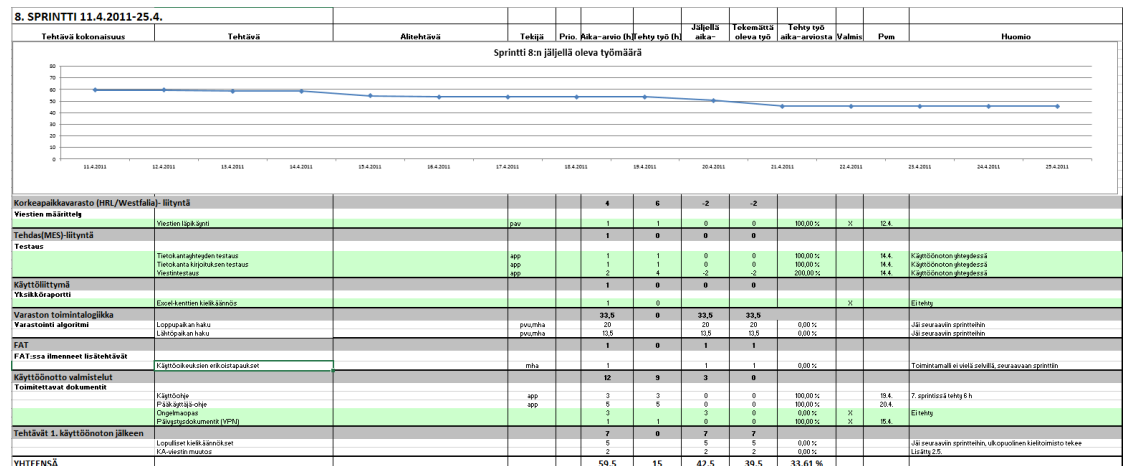
Julkaisun edistymiskäyrä toteutettiin Excelin kaavio-työkalulla suoraan tuotteen työlistasta. Edistymiskäyrä päivittyi lähes automaattisesti aina kun työlistaa päivitettiin. Edistymiskäyrä oli oiva työkalu mistä näki nopeasti projektin edistymisen. Kuviossa 17 on kuvattu meidän julkaisun edistymiskäyrä.



KUVIO 17. Julkaisun edistymiskäyrä.

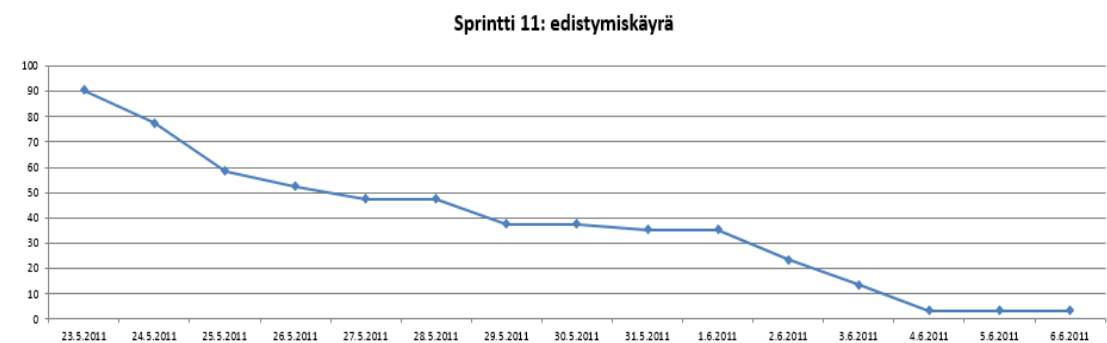
Sprintin työlistaan listasimme sprinttiin valitut tehtävät, tähän listaukseen lisäsimme tehtävälle myös vastuuhenkilön. Työlistaa päivitettiin sprintin edetessä ja saatoimme

tietyissä tilanteissa lisätä tehtäviä kesken sprintin tähän työlialle. Sprintin työlialta oli helppo valita tehtäviä Scrum-tiimin jäsenille päiväpalaverissa ja käytännössä tämä oli se dokumentti mitä käsiteltiin aina sprintin palaverissa. Kuviossa 18 on tässä projektissa käytetty sprintin työlialta.



KUVIO 18. Sprintin työlialta.

Sprintin edistymiskäyrä toteutettiin kuten julkaisun edistymiskäyrä eli teimme Excel-kaavion sprintin työlialta. Kaavio päivittyi lähes automaattisesti samalla kun sprintin työlialta päivittyi. Kaaviosta näki nopeasti sprintin edistymisen. Kuviossa 19 on malli käyttämästämme sprintin edistymiskäyrästä.



KUVIO 19. Sprintin edistymiskäyrä.

Tarvittavat päivitykset näihin dokumentteihin teki pääsääntöisesti Scrum-mestari, mutta myös muut tiimiläiset tekivät päivityksiä. Ideaali tilanne olisi, jos tiimin jäsenet päivittäisivät itse tekemänsä tehtävät sprintin työlialta ja nämä päivitykset menisivät automaattisesti sekä molempiin edistymiskäyriin, että tuotteen työlialta. Vielä

tämän projektin dokumenteissa tarvittiin Scrum-mestarin työtä, jotta kaikki dokumentit saatiin ajan tasalle.

6.1.4 Scrumin muut hyödyt Metsysille

Metsysillä ei ollut projekteissa käytössä tuntiseurantaa. Metsysin toteuttamaan Scrum-menetelmään kuului seurata tehtävien edistymistä tuntien tarkkuudella ja kaikille tehtäville annettiin tuntiarvio. Eli Scrumin sivutuotteena saatiin projektiin tuntiseuranta. Pitää kuitenkin huomioida, että Scrumin tuntiseuranta ei ole menetelmän itsetarkoitus vaan yksi osa koko menetelmää. Eikä seurantaa voida pitää 100 % luotettavana vaan enemmänkin suuntaa-antavana. Hyöty Metsysille oli kuitenkin merkittävä, koska tämän projektin jälkeen oli helpompi ymmärtää, kuinka paljon aikaa ja työtä menee projektien tehtävien toteuttamiseen. Yleisesti, myös tässä projektissa, tuntiarviot ovat olleet aika reilusti alakantissa. Ehkä usein huomioidaan vain se nettoaika mikä menee työn konkreettiseen tekemiseen, mutta unohdetaan aika mikä kuluu työhön epäsuorasti.

Useasti projektien yksi suurimmista ongelmista on kommunikaation riittämättömyys projektin jäsenten kesken. Scrumia käytettäessä tulee väkisinkin kommunikoida säännöllisesti projektiryhmän kesken, joten koko projektista tulee huomattavasti läpinäkyvämpi. Läpinäkyvyys edesauttaa projektin hallinnoitua ja estää mahdollisten turhien ja päällekkäisten tehtävien tekemisen. Lisäksi läpinäkyvyys antaa hyvän mahdollisuuden mm. yrityksen johdolle seurata projektin edistymistä.

6.2 Scrum-menetelmän haasteet ja ongelmat

Scrum-menetelmä on kattava projektinhallinnan menetelmä, joten suoraan siihen siirtyminen on iso muutos koko projektiorganisaatiolle. Varsinkin pilottiprojektissa, kun kaikki käytännöt ja menetelmät ovat kaikille uusia, on haastavaa saada Scrum-projekti etenemään sulavasti. Lisäksi ei voida olla huomioimatta muutosvastarintaa, mitä esiintyy aina, kun siirrytään vanhasta tutusta menetelmästä uuteen tapaan toimia. Nämä edellä mainitut haasteet olivat projektiryhmällä tiedossa jo ennen projektin aloittamista ja osin tästä syystä sovimme, että kevennämme Metsysin käyttämää Scrum-menetelmää joiltakin osin. Emme järjestäneet päiväpalaveria päivittäin ja yhdistimme Sprintin katselmuksen - ja Sprintin jälkitarkastelu-palaverit yhdeksi

palaveriksi sprintin loppuun. Jos kaikki palaverit olisi järjestetty Scrum-menetelmän mukaisesti, olisi 10 päivän sprintistä mennyt vähintään 2 päivää palavereihin, mikä on suhteellisen suuri osuus sprintin työajasta. Jälkeenpäin ajatellen päiväpalaverit olisi kenties ollut parempi järjestää päivittäin. Kun palaverit järjestettiin muutaman päivän välein, oli hankalaa saada palaverit pysymään 15 minuutin mittaisina, koska kerrottavaa oli kaikille tiimiläisille kertynyt useamman päivän töistä. Yleensäkin Scrum-mestarin pitää olla tiukempi päiväpalavereissa, jotta keskustelu pysyy Scrumissa sovituissa kehyksissä, eikä lähde rönstyilemään palaveriin kuulumattomiin asioihin.

Omat ongelmat aiheuttivat myös tähän Scrum-projektiin tehdyt dokumentit ja lähinnä niiden päivittäminen ja ylläpitäminen. Projektin edistyessä tuotteen työlistaan tuli uusia tehtäviä, tehtävien aika-arvioihin saattoi tulla muutoksia, osa tehtävistä jätettiin kokonaan tekemättä ja osalla tehtävistä aika-arvio joko ylittyi tai alittui. Kaikki nämä edellä mainitut muutokset vaikuttivat tuotteen työlistan dataan ja sitä kautta julkaisun edistymiskäyrään. Meidän käyttämissä tuotteen – ja sprintin työlistoissa oli se ongelma, että ne eivät eritelleet tunteja, jotka ylittivät tai alittivat aika-arvion eli, jos saimme sprintissä kaikki tehtävät suoritettua alle aika-arvion, näkyi edistymiskäyrällä edelleen tekemättömiä töitä. Kaiken kaikkiaan dokumenttien ylläpitämiseen meni Scrum-mestariolta useampi tunti joka viikko. Tuleviin Scrum-projekteihin olisi hyvä kehittää Excel-dokumentit helpommin hallittavaan muotoon tai siirtyä jopa kokonaan tietokantapohjaiseen järjestelmään.

6.3 Scrumin toimivuus Metsysin projektissa

Metsysin ohjelmistoprojektit ovat kestoltaan noin puoli vuotta, josta ensimmäiset 3-4 kuukautta tehdään ohjelmointityötä projektin alussa tehdyn speksin pohjalta. Projektin 2-3 viimeistä kuukautta ovat intensiivisempää tekemistä yhdessä projektin muiden osapuolten kanssa. Tämän vaiheen lopussa tehdään järjestelmän käyttöönotto, mikä on koko projektin kiireellisintä aikaa.

Scrum-menetelmä toimi hyvin tämän projektin ensimmäisten 6 sprintin aikana, kun kehitystyötä suoritettiin Scrum-tiimin voimin samalla toimistolla ja projektin

eteneminen oli stabiilia. Tässä vaiheessa speksin muutoksiin pystyttiin reagoimaan hallitusti Scrum-menetelmän käytäntöjä hyödyntäen.

Haasteita Scrumin prosesseihin tuli, kun projekti siirtyi käyttöönottovaiheeseen. Scrum prosessissahan kaikki uudet tehtävät lisätään ensin tuotteen työlistalle, josta Scrum-mestari pystyy lisäämään tehtävät sprintin työlistalle, josta tehtäviä voidaan alkaa suorittamaan tiimiläisten toimesta. Käyttöönotto on Metsysin projekteissa aina intensiivinen johtuen loppuasiakkaiden tarpeesta saada järjestelmä käyttöön mahdollisimman nopeasti ja häiritsemättä yleensä käynnissä olevaa tuotantoa. Edellä kuvatuissa olosuhteissa ei ole varaa yhtään pitkittää virheiden korjausta tai muutosten tekemistä, vaan nämä pitää yleensä tehdä välittömästi. Tällöin ei ole järkevää, eikä aina käytännössä mahdollista edetä Scrumin prosessien mukaan, vaan tehtävät pitää tehdä välittömästi ohi Scrum prosessien. Lisäksi käyttöönotossa mukana oleva Metsysin asiantuntija on yleensä ohjelmointitaitoinen ja pystyy tekemään tarvittavat muutokset ohjelmistoon suoraan, ilman Scrum-tiimin apua. Tällöin ei ole järkevää kierrättää tehtäviä työlistojen kautta. Tässäkin projektissa tehtiin joitakin kiireellisiä muutostöitä suoraan ohi Scrumin prosessien, työt lisättiin työlistoihin jälkikäteen ja merkittiin suoritetuiksi.

Haasteita aiheutti myös projektiryhmän pienuus ja se että osa tiimiläisistä työskenteli samanaikaisesti muissa Metsysin projekteissa eli he olivat välillä poissa Scrum-tiimin vahvuudesta. Parhaimmillaan Scrum-tiimissä oli kuusi henkilöä, mutta toisinaan oli ajanjaksoja, että projektissa työskenteli vain kaksi henkilöä. Projektin henkilömäärä saattoi vaihdella kesken sprintin, josta seurasi ongelmia saada valmiiksi kaikki sprintille valitut tehtävät. Projektihenkilöiden siirtyminen projektista toiseen ei johtunut Scrum-menetelmästä, vaan olivat pakonsanelemia muutoksia, joita ainakin pienissä yrityksessä useasti joudutaan tekemään. Jos projektin tiimiläinen on hetkellisesti poissa Scrum-projektista, ei hänen ole välttämättä mahdollista osallistua kaikkiin Scrumin palavereihin ja tällöin hän menetetään hyödyn Scrumin mahdollistavasta läpinäkyvyydestä.

6.4 Vaikutus laatuun

Scrumin vaikutusta ohjelmistotuotteen laatuun on vaikea mitata yhden projektin perusteella. Uskon, että vaikutus laatuun tämän projektin osalta on tullut paremman kommunikoinnin ja ohjatun kehitystyön kautta. Metsysin käyttöönotetut järjestelmät ovat aina olleet luotettavia ja laadukkaita, mutta siihen prosessiin, miten laadukkaaseen järjestelmään päästään voidaan vaikuttaa paremmilla toimintatavoilla. Tässä näen Scrumilla olevan suuri mahdollisuus tulevilla projekteilla.

Järjestelmän testauksessa Scrum-menetelmästä voidaan saada suurta hyötyä ja tätä kautta voidaan vaikuttaa järjestelmän laatuun. Wattensin projektissa emme hyödyntäneet Scrumin tuomaa etua tarpeeksi hyvin testauksessa. Järjestelmää olisi voitu testata hallitusti lisäämällä ns. testi caset sprintin työlistalle, tällöin testaaja olisi voinut päiväpalavereissa kertoa muille tiimiläisille, kuinka tehtävät eli testi caset edistyvät ja onko ongelmia edistymisessä. Tällöin myös testaaminen olisi läpinäkyvää läpi projektiryhmän. Tässä on vielä parantamisen varaa tulevilla Scrum-projekteilla.

6.5 Vaikutus kustannuksiin

Tämä pilottiprojekti ei suorasti vaikuttanut positiivisesti projektin kustannuksiin, mutta jos Scrumin prosessit saadaan hiottua järkeviksi, on vaikutus nähtävissä projektien läpimenoaikojen pienenemisellä. Tällä on suora vaikutus kustannuksiin. Toisaalta Scrum-organisaatio on hieman suurempi, kuin perinteisen ohjelmistoprojektin, joten taloudellisessa mielessä hyödyn saaminen on ainakin alussa haastavaa.

Taloudellisessa mielessä olisi kannattavaa, jos pystyttäisiin yhdistämään useampi projekti yhdeksi Scrum-projektiksi. Tällöin yhdellä Scrum-organisaatiolla pystyttäisiin hallinnoimaan samanaikaisesti useita projekteja.

7 POHDINTA

Tämän työn tekeminen oli allekirjoittaneelle erittäin antoisa kokemus. Projekti lähti liikkeelle työtoimeksiantajan ehdotuksesta kehittää Metsysin projektinhallintaa ns.

ketteriin menetelmiin kuuluvalla Scrum-menetelmällä. Allekirjoittaneella ei ollut aikaisempaa kokemusta Scrum-menetelmästä, joten ensimmäinen tehtävä oli tutustua kyseiseen menetelmään. Kirjallisuutta Scrumin ympäriltä löytyy hyvin ja myös netistä löytyy paljon kirjoitettua tietoa. Oppimista oli paljon ja menetelmän opiskelu jatkui läpi pilottiprojektin. Kun allekirjoittaneella oli tarpeeksi tietoa Scrum-menetelmästä tuli vuoroon menetelmän kouluttaminen Metsysin noin 10 hengen henkilöstölle. Yhdessä henkilöstön kanssa pohdimme, miten saamme Scrum-menetelmän sovitetuksi Metsysin toimintaympäristöön sopivaksi. Jouduimme muokkaamaan joitakin Scrumin toimintaperiaatteita, johtuen projektiryhmän pienuudesta ja siitä että osa tiimiläisistä työskenteli samanaikaisesti useammassa projektissa.

Pilottiprojektiksi valikoitui ajankohdallisesti sopivasti alkanut projekti. Projekti oli käynnistynyt jo loppuvuodesta 2010 ja Scrum liitettiin tähän projektiin hieman tämän jälkeen. Allekirjoittanut toimi projektissa Scrum-mestarina ja Scrum-tiimin muodostivat Metsysin ohjelmistokehittäjät. Scrumin osalta projekti alkoi tuotteen työlistan tekemisellä. Tämän listan pohjalta projekti alettiin tekemään sprintti kerrallaan. Alun perin oli suunniteltu viedä projekti lävitse kuudessa sprintissä, mutta lopulta sprinttejä järjestettiin 11.

Scrumin osalta projekti meni lävitse, kuten olimme projektin alussa sopineet. Järjestimme jokaisen sprintin alussa sprintin aloituspalaverin missä kävimme lävitse tulevan kahden viikon työt. Nämä työt lisättiin sprintin työlistalle. päiväpalavereja emme pitäneet päivittäin, mutta säännöllisesti kuitenkin. Sprintin jälkeen pidimme samalla kertaa Sprintin katselmuksen, että Sprintin jälkitarkastelun. Jo näillä palavereilla saimme läpinäkyvyyttä projektiin ja tieto kulki projektin sisällä kiitettävästi.

Kuten aikaisemmassa tekstissä on mainittu, vaihtui Scrum-tiimin vahvuus projektin aikana, tämä aiheutti ongelmia Scrumin toiminnalle. Ehdottomasti olisi parempi, että Scrum-tiimi olisi stabiili läpi projektin. Tällöin tiimiläiset voisivat luottaa, että kaikki tiedävät tarkasti projektin vaiheet ja ovat selvillä mitä on tapahtunut, mitä tullaan tekemään ja miksi. Turhat epäselvyydet ja päällekkäiset työt projektin sisällä jäisivät minimiin ja olisihan se myös tekijän näkökulmasta paras mahdollinen tilanne, että pystyisi keskittymään yhteen projektiin kerrallaan. Toisekseen sopivan määrän töiden valitseminen Sprintin työlistalle on haastavaa, jos ei tiedä kuinka paljon on

käytettävissä tiimiläisten resursseja sprintin aikana. Jos resursseja on sprintissä vähemmän kuin on uskottu sprintin alussa, jää todennäköisesti osa sprintin töistä suorittamatta.

Tiedossa on kuitenkin myös se, että pienessä firmassa ei ole aina mahdollista jakaa töitä siten, että yksi tekijä tekee yhtä projektia kerrallaan. Useasti Metsysillä suoritetaan samanaikaisesti 2-3 projektia ja jo erityyppinen ohjaa asiantuntijoita sekaantumaan näihin rinnakkaisiin projekteihin. Tämä realiteetti tiedostaen, on erittäin varteenotettava vaihtoehto tutkia mahdollisuutta suorittaa useita projekteja yhden Scrum-projektin alla. Tällöin jokaisen projektin jokainen tekijä olisi osa yhteistä Scrum-tiimiä, joka kokoontuisi palavereihin Scrum-menetelmän mukaisesti. Palavereissa käsiteltäisiin hallitusti kaikki meneillään olevat projektit. Tällöin tieto olisi läpinäkyvää yli projektirajojen ja tiimiläisten olisi helpompi siirtyä projektista toiseen. Tällä mallilla Scrum-organisaatio olisi huomattavasti kevyempi, verrattuna että jokaisella projektilla olisi oma projektiorganisaatio. Ongelmana tässä olisi palaverien keston pidentyminen ja Scrumiin kuuluvien dokumenttien ylläpitäminen. Palaverien läpiviemiseen pitäisi kehittää sujuva malli, jossa ajankäyttö olisi tehokasta ja keskittyttäisiin vain merkittäviin asioihin, tässä Scrum-mestarilla olisi suuri rooli. Dokumenttien ylläpitäminen pitäisi olla mahdollisimman vaivatonta ja tähän perinteinen Excel-taulukko ei ole paras mahdollinen työkalu. Parhaiten tähän tarkoitukseen sopisi Scrum-projektinhallinta sovellus, missä kaikki tarvittava data säilöittäisiin ja prosessoitaisiin tietokannassa.

Tämä malli, missä yhdellä Scrumilla hallitaan useampaa projektia, ei ole tietenkään Scrumin mukainen toimintatapa. Sen toimivuutta ei ole missään todettu ja siinä menetetään osa Scrumin tuomista hyödyistä. Mutta allekirjoittaneen mielestä ehdottomasti miettimisen arvoinen asia ja kenties tulevaisuudessa tätä mallia voidaan aluksi testata sopivissa pilottiprojekteissa.

Toinen harkinnan arvoinen tapa millä Metsysin projektit voitaisiin viedä lävitse nykyistä hallitummin, olisi tehdä yksi projekti kerrallaan, koko henkilöstön voimin ja Scrumin ohjaamana. Tällä mallilla välttyttäisiin projektista toiseen siirtymiltä ja jokainen kehittäjä pystyisi keskittymään yhteen projektiin kerrallaan. Näin myös jokainen tiimiläinen saisi kosketuspintaa jokaiseen projektiin mistä olisi varmasti hyötyä projektien ylläpitoon. 400 tunnin projekti pystyttäisiin teoriassa tekemään

neljän ohjelmistokehittäjän voimin kahdessa kahden viikon sprintissä. Tämän jälkeen tiimi olisi valmis tekemään seuraavaa projektia. Tietenkään nämä eivät mene näin lineaarisesti, mutta teoriassa myös tämä malli voisi olla toimiva Metsysin käytössä.

Opinnäytetyöhön liittyi kolmiosainen kysely projektinhallinnasta Metsysin henkilöstölle. Kyselyihin tulleiden vastausten perusteella voidaan sanoa, että Metsysin henkilöstö toivoo hallitumpaa projektinhallintaa. Hallitulta projektinhallinnalta toivottiin ensisijaisesti projektipäällikön nimeämistä, töiden organisointia, työlistausten tekemistä, projektin parempaa läpinäkyvyyttä, organisoidumpaa testausta ja projektin kattavaa läpikäymistä projektin jälkeen.

Tätäkin pohjaa vasten Scrum-menetelmän käyttöönottoaminen Metsysin projektinhallintaan olisi perusteltua. Scrumilla pystyttäisiin parantamaan lähes jokaista kyselyissä esille nousutta ongelmaa. Toki nuo ongelmat johtuivat pitkälti siitä, että käytössä ei ole minkäänlaista projektinhallintaa ja ongelmat pystyttäisiin, ainakin osittain, korjaamaan panostamalla projektien sujuvaan läpiviemiseen.

Lopuksi yhteenvetona kyselyjen vastausten perusteella ja Metsysin projekteissa mukana olleena, voin sanoa, että tarve ohjatulle projektinhallinnalle on olemassa. Se, toteutetaanko projekti Scrumilla vai jollain muulla projektinhallinnan menetelmällä on tavallaan toissijaista, tärkeintä on, että käytössä on joku menetelmä. Todettakoon kuitenkin, että Scrum soveltuisi oikein käytettynä erittäin hyvin Metsysin projektinhallinta menetelmäksi.

LÄHTEET

Abrahamsson Pekka, Warsta Juhani, Siponen Mikko T, Ronkainen Jussi. 2003. New Directions on Agile Methods: A Comparative Analysis.

Asteis David, Granville Miller, Novak Miroslav. 2002, A Practical Guide to Extreme Programming. United States: (NJ) Prentice Hall.

Brooks Frederick. 1986, No Silver Bullet.

Cockburn Alistair. 2007, Agile Software Development: The Cooperative Game (2nd Edition. United States: Addison-Wesley Professional, Inc.

Haikala Ilkka ja Märijärvi Jukka. 2004, Ohjelmistotuotanto. 10. painos. Karisto Oy. Hämeenlinna 2004.

Highsmith James. 2000, Adaptive Software Development – A Collaborative Approach to Managing Complex Systems. United States: Dorset House.

Leffingwell Dean. 2007. Scaling software agility: Best Practices for Large Enterprises. United States: Addison-Wesley Professional, Inc.

Lehtimäki Timo. 2006. Ohjelmistoprojektit käytännössä. 1. painos. Gummerus Kirjapaino Oy, Jyväskylä 2006.

Schuh Peter. 2005. Integrating Agile Development in the Real World. 1. painos. United States: Charles River Media, Inc. 2005.

Schwaber Ken. 2004. Agile Project Management with Scrum. United States: Microsoft Press.

Schwaber Ken ja Beedle Mike. 2002. Agile Software Development with Scrum. United States: Upper Saddle River (NJ), Prentice Hall.

Schwaber Ken ja Sutherland Jeff. 2013. Scrum Guide. Viitattu 27.4.2014.
<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100>.

Sininen Meteoriitti: Ketteryys haltuun. 2011. Viitattu 27.4.2014.
<http://www.meteoriitti.com/Artikkelisarjat/Ketteryys-haltuun/Ketteryys-haltuun-Scrum-pahkinankuoressa/>

Wikipedia. 2014. Ketterä ohjelmistokehitys. Viitattu 27.4.2014.
http://fi.wikipedia.org/wiki/Ketter%C3%A4_ohjelmistokehitys

LIITTEET

Liite 1. Agile Manifesti

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The Twelve Principles of Agile Software.

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Liite 2. Ensimmäinen kysely

KYSELYLOMAKE

Kysymyksillä pyritään saamaan lähtökohtatiedot yrityksen projektinhallinnasta. Saatuja tietoja käytetään opinnäytetyön lopullisten tulosten laatimiseen ja yrityksen projektinhallinnan parantamiseen. Kysymykset ovat tarkoituksella jätetty laajoiksi ja avoimiksi, joten voitte kirjoittaa vastaukseen kaikki mitä mieleenne tulee nykyisestä projektinhallinnasta. Tiedot ovat erittäin tärkeitä työni onnistumisen kannalta ja myös yrityksen projektinhallinnan kehittämisen vuoksi, joten toivottavasti vastaatte kyselyyn. Palauttakaa täytetty dokumentti joko sähköisesti tai kirjallisesti minulle viimeistään 25.11.2010. Tuloksia analysoidaan opinnäytetyössä kokonaisuutena, eikä yksittäisinä vastauksina. Kaikki tiedot ovat 100 % luottamuksellisia!

1. Mitä hyviä ominaisuuksia nykyisessä projektinhallinnassa on?

2. Mitä kehitettäviä (huonoja) osa-alueita nykyisessä projektinhallinnassa on?

Liite 3. Toinen kysely (ennen Scrum-projektia)

KYSELYLOMAKE 2

Moi taas!

Kiitokset kaikille hyvistä vastauksista edellisiin kysymyksiini. Nyt kyselen vähän tarkemmin muutamia kysymyksiä. Vastatkaa asteikolla 0-10 (**0**=Täysin erimieltä, **5**=jokseenkin samaa mieltä ja **10**= täysin samaa mieltä). **Lihavoikaa valitsemanne numero!**

Saatuja tietoja käytetään opinnäytetyön lopullisten tulosten laatimiseen ja yrityksen projektinhallinnan parantamiseen. Tiedot ovat erittäin tärkeitä työni onnistumisen kannalta ja myös yrityksen projektinhallinnan kehittämisen vuoksi, joten toivottavasti vastaatte kyselyyn. Jättäkää vastauksissa huomioimatta meneillään oleva **Wattensin-projekti**. Kyselen samat kysymykset keväällä uudelleen. Palauttakaa täytetty dokumentti sähköisesti minulle viimeistään perjantaina 4.2.2011. Tuloksia analysoidaan opinnäytetyössä kokonaisuutena, eikä yksittäisinä vastauksina. Kaikki tiedot ovat 100 % luottamuksellisia! Kiitokset etukäteen!

Vastatkaa asteikolla 0-10 (**0**=Täysin erimieltä, **5**=jokseenkin samaa mieltä ja **10**= täysin samaa mieltä). **Lihavoikaa valitsemanne numero!**

1. Metsysin projektien joustavat työskentelytavat ovat toimivia.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

2. Alkavat projektit on speksattu kattavasti.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

3. Kesken projektin on helppo reagoida speksin muutoksiin.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

4. Projektien työlista on selkeä ja ajan tasalla.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

5. Metsysin projektit ovat läpinäkyviä (avoimia, julkisia, kaikilla projektin osallistujilla on mahdollisuus päästä kaikkiin tarpeellisiin tietoihin)

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

6. Olet selvillä tekemäsi projektin valmiusasteesta.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

7. Olet tietoinen missä vaiheessa muut projektit ovat.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

8. Olet tietoinen mitä työtehtäviä työkaverisi tekevät.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

9. Vastuualueet ovat projekteissa selvillä.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

10. Töiden organisoiminen on toimivaa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

11. Töiden priorisoiminen on toimivaa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

12. Ohjelmien dokumentoiminen on kunnossa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

13. Suoritetut projektit käydään kattavasti läpi projektin jälkeen.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

14. Nykyinen projektinhallinta on toimiva.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

15. Edellisiä projekteja pystytään hyödyntämään alkavissa projekteissa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

16. Edellisiä projekteja hyödynnetään hyvin alkavissa projekteissa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

17. Testaus projekteissa on johdonmukaista.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

18. Testaus projekteissa on riittävää.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

19. Tiedät milloin projekti on päättynyt.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

20. Kokopäiväinen projektipäällikkö selventäisi projektien läpivientä.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

Liite 4. Kolmas kysely (Scrum-projektin jälkeen)

KYSELYLOMAKE 3

Moi taas!

Kiitokset kaikille hyvistä vastauksista edellisiin kysymyksiini. Nyt kyselen vähän tarkemmin muutamia kysymyksiä. Vastatkaa asteikolla 0-10 (0=Täysin erimielistä, 5=jokseenkin samaa mieltä ja 10= täysin samaa mieltä). **Lihavoikaa valitsemanne numero!**

Saatuja tietoja käytetään opinnäytetyön lopullisten tulosten laatimiseen ja yrityksen projektinhallinnan parantamiseen. Tiedot ovat erittäin tärkeitä työni onnistumisen kannalta ja myös yrityksen projektinhallinnan kehittämisen vuoksi, joten toivottavasti vastaatte kyselyyn. Painottakaa vastauksissa meneillään olevaa **Wattensin-projektia**. Kysyin samat kysymykset alkuvuonna ennen Scrumin käyttöönottoa. Palauttakaa täytetty dokumentti sähköisesti minulle viimeistään perjantaina 24.6.2011. Tuloksia analysoidaan opinnäytetyössä kokonaisuutena, eikä yksittäisinä vastauksina. Kaikki tiedot ovat 100 % luottamuksellisia! Kiitokset etukäteen!

Vastatkaa asteikolla 0-10 (0=Täysin erimielistä, 5=jokseenkin samaa mieltä ja 10= täysin samaa mieltä). **Lihavoikaa valitsemanne numero!**

1. Metsysin projektien joustavat työskentelytavat ovat toimivia.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

2. Alkavat projektit on speksattu kattavasti.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

3. Kesken projektin on helppo reagoida speksin muutoksiin.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

4. Projektien työlista on selkeä ja ajan tasalla.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

5. Metsysin projektit ovat läpinäkyviä (avoimia, julkisia, kaikilla projektin osallistujilla on mahdollisuus päästä kaikkiin tarpeellisiin tietoihin)

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

6. Olet selvillä tekemäsi projektin valmiusasteesta.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

7. Olet tietoinen missä vaiheessa muut projektit ovat.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

8. Olet tietoinen mitä työtehtäviä työkaverisi tekevät.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

9. Vastuualueet ovat projekteissa selvillä.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

10. Töiden organisoiminen on toimivaa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

11. Töiden priorisoiminen on toimivaa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

12. Ohjelmien dokumentoiminen on kunnossa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

13. Suoritetut projektit käydään kattavasti läpi projektin jälkeen.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

14. Nykyinen projektinhallinta on toimiva.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

15. Edellisiä projekteja pystytään hyödyntämään alkavissa projekteissa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

16. Edellisiä projekteja hyödynnetään hyvin alkavissa projekteissa.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

17. Testaus projekteissa on johdonmukaista.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

18. Testaus projekteissa on riittävää.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

19. Tiedät milloin projekti on päättynyt.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

20. Kokopäiväinen projektipäällikkö selventäisi projektien läpivientä.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----